Massachusetts Institute of Technology

Project MAC Progress Report IV July 1966 to July 1967

281

Massachusetts Institute of Technology
Project MAC
545 Technology Square
Cambridge, Massachusetts
02139

Work reported herein was supported by Project MAC, an M.I.T. research program sponsored by the Advanced Research Projects Agency, Department of Defense, under Office of Naval Research Contract Number Nonr-4102(01). The primary support for some of this work came from the M.I.T. departments and laboratories participating in Project MAC, whose research programs are, in turn, sponsored by various government and private agencies. This support is acknowledged by specific mention of agency and contract number in the appropriate sections.

Government contractors may obtain copies of this Progress Report, and all Project MAC Technical Reports listed in Appendix D, from the Defense Supply Agency, Cameron Station, Alexandria, Virginia 22314. Response will be expedited if requests are submitted on DDC Form 1, copies of which are available from the office in your activity which has been established as the focal point for requesting DDC services.

Other U.S. citizens and organizations may obtain copies of this report from the Clearinghouse for Federal Scientific and Technical Information (CFSTI), Sills Building, 5285 Port Royal Road, Springfield, Virginia 22151.

BLANK PAGE

ADMINISTRATION

ARTIFICIAL INTELLIGENCE

BIOLOGY DEPARTMENT

CIVIL ENGINEERING DEPARTMENT

COMPUTATION STRUCTURES

COMPUTER SYSTEM RESEARCH

ELECTRONIC SYSTEMS LABORATORY

HARVARD UNIVERSITY

MAN-MACHINE COMMUNICATION

PROGRAMMING LINGUISTICS

RESEARCH LABORATORY OF ELECTRONICS

SCHOOL OF ENGINEERING

SCHOOL OF HUMANITIES AND SOCIAL SCIENCES

SCHOOL OF SCIENCE

SLOAN SCHOOL OF MANAGEMENT

TECHNICAL INFORMATION PROGRAM

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

LIST OF ILLUSTRATIONS

PERSONNEL

FROM JULY 1, 1966 TO JULY 1, 1967

Administration

Prof. R. M. Fano
- Director

R. G. Mills
- Assistant Director

Committee On Computation

Prof. P. M. Morse, Chairman
Prof. R. M. Fano
Prof. R. I. Hulsizer, Jr.
Prof. W. B. Kehl
Prof. C. L. Miller
Prof. C. F. J. Overhage
Dean W. F. Pounds
Prof. L. D. Smullin
Dean J. B. Wiesner

Academic Staff

S. S. Alexander
A. E. Amstutz
C. A. Berg
G. D. Bernard
A. Bers
J. M. Beshers
K. Biemann
J. M. Biggs
W. L. Black
M. Blum
F. Bonilla
S. C. Brown
D. C. Carroll
J. G. Charney
S. A. Coons
P. H. Cootner
F. J. Corbató
W. Crowston
J. DaSilva
J. B. Dennis
M. L. Dertouzos
A. G. H. Dietz
J. J. Donovan
A. Evans, Jr.
R. M. Fano
D. E. Farrar
R. R. Fenichel
W. R. Fey
J. W. Forrester
E. L. Glaser
L. A. Gould
M. Greenberger
M. Haire
M. Halle
K. F. Hansen
H. A. Haus
W. L. Henke
F. C. Hennie, III
A. M. Hershdorfer
D. M. Holland
L. N. Howard
D. A. Huffman
M. M. Jones
R. Kaplow
D. Klatt

Academic Staff (continued)

Z. Kohavi
D. A. Kolb
E. Kuh
F. F. Lee
H. B. Lee, Jr.
D. Lerner
C. Levinthal
J. C. R. Licklider
J. D. C. Little
C. L. Liu
R. D. Logcher
P. Mandel
D. G. Marquis
D. F. Martin
S. J. Mason
R. I. Matales
F. A. McClintock
H. S. Mickley
M. L. Minsky
C. F. J. Overhage
H. M. Paynter
P. L. Penfield, Jr.
F. E. Perkins
G. H. Poensgen
I. de S. Pool
J. F. Reintjes
E. B. Roberts
D. Roos
R. C. Rosenberg
J. H. Saltzer
C. E. Shannon
T. B. Sheridan
L. D. Smullin
K. N. Stevens
G. M. Sturman
A. K. Susskind
R. D. Thornton
A. Toomre
D. E. Troxel
J. Waelbroeck
J. Weizenbaum
L. S. White
R. V. Whitman
J. M. Wozencraft
K. Zander
H. J. Zimmermann

Research Associates, Instructors and Lecturers

G. W. Angell
C. D. Barry
L. N. Beckreck
D. G. Bobrow
J. W. Brackett
A. J. Ferrante
A. T. Foster
G. A. Gorry, Jr.
P. J. Landin
W. H. Linder
D. N. Ness
S. A. Papert
R. P. Parmelee
W. M. Pecknold, III
A. L. Pugh, III
H. M. Schneider
C. R. Sprague
A. Stoper
J. M. Sussman
C. V. Swanson
R. A. Walter

Non-Academic Research Staff

R. J. Abbott	U. F. Gronemann	R. C. Nelson
E. C. Bartels	J. A. Gunn	W. R. Noftsker
M. D. Beeler	E. G. Guttmann	L. P. Odland
W. Bennett	D. R. Haring	C. D. Olmsted
G. L. Benton	J. Harwitt	M. Padlipsky
R. J. Bigelow	W. H. Henneman	J. A. C. Parisot
H. E. Brammer	F. B. Hills	M. M. Pennell
M. L. Cabral	P. Himot	R. B. Polansky
E. J. Campbell	J. T. Holloway	R. Rappaport
J. P. Casey	E. Issacs	M. Richards
T. B. Cheek	E. R. Jensen	D. M. Ritchie
G. F. Clancy	B. A. Johanson	E. C. River
R. W. Cornew	J. C. Jordan	S. L. Rosenbaum
C. A. Cushing	M. J. Kahne	D. T. Ross
M. M. Daggett	M. M. Kessler	K. D. Rude
R. C. Daley	P. Kugel	A. J. Saltalamacchia
R. L. Daniels	R. L. Kusik	P. Samson
H. S. Davis	R. Leopold	R. A. Sayers
T. F. Dempsy	H. D. Levin	P. M. Sheehan
M. E. Denofsky	A. Luehrmann	P. Smith
R. A. DePrisco	C. Marceau	M. V. Solomita
E. A. Dole	R. S. Marcus	W. A. Solomon
M. Eisner	K. J. Martin	J. W. Spall
B. J. Fay	W. D. Mathews	J. E. Spencer
C. G. Feldmann	E. M. Mattison	N. C. Spencer
C. Garman	S. D. McIntosh	G. Spielman
R. V. Goodman	E. F. Miller	M. J. Spier
R. W. Gosper	R. G. Mills	R. L. Stetson
R. M. Graham	N. I. Morris	R. H. Stotz

Non-Academic Research Staff (continued)

W. D. Stratton
R. L. Taggart
E. F. Taylor
M. K. Thompson
D. E. Thornhill
O. J. Tretiak
T. H. Van Vleck
A. Vezza
G. Voyat
D. B. Wagner
J. F. Walsh
J. E. Ward
S. A. Ward
T. Warner
D. R. Widrig, Jr.
B. L. Wolman

Resident Guests

S. I. Ackley	System Development Corp.
N. Adleman	System Development Corp.
R. H. Anderson	Harvard University
P. Anesse	U. S. Navy
D. Barovich	I.B.M. Corporation
A. J. Berger	Boeing Company
W. L. Johnson	Ford Motor Company
J. H. Jones	McDonnell Aircraft Corp.
F. Luccio	Olivetti and Company
R. A. Meyer	I.B.M. Corporation
A. K. Mills	Dow Chemical Company
J. H. Porter	Chevron Research Company
R. Silver	MITRE Corporation
I. Wenger	Raytheon SIS
R. B. Wise	IIT Research Institute
S. Zurnaciyan	Northrop

Non-M. I. T. Users

V. Guiliano	Arthur D. Little, Inc.
R. Taylor	ARPA - Department of Defense
A. G. Oettinger	Harvard University
A. Ruyle	Harvard University
I. Sutherland	Harvard University
K. Winiecki	Harvard University

Research Assistants

R. Baecker	A. Guzmán-Arenas	S. S. Patil
T. J. Barkalow	E. J. Hall	D. N. Perkins, Jr.
E. R. Banks	P. G. Hebalkar	T. Pitidis-Poutous
R. R. Bartsch	C. P. Hensler	I. Plotkin
P. Berman	B. J. Huffman	P. Raup
W. T. Beyer	T. J. R. Johnson	J. E. Rodriguez
R. J. Bobrow	T. L. Jones	H. L. Selesnick
A. A. Bushkin	F. Kamijo	J. J. Sierakowski
T. Carroll	W. R. Kampe	D. R. Slutz
S. D. Chan	L. S. Kanodia	C. E. Speck
P. Choong	A. R. Kessler	L. C. Teague, Jr.
C. Chryssostomidis	L. J. Krakauer	C. W. Therrien
S. D. Crocker	J. F. Kramer	C. C. Tillman, Jr.
R. Crossley	B. R. Kusse	M. A. Trozzi
R. Diephuis	R. B. Lapin	P. M. Walker
D. S. Evans	H. F. Ledgard	D. L. Waltz
G. Falk	J. L. Linderman	M. Weinberger
R. C. Gammill	F. L. Luconi	B. D. Wessler
S. Geffner	H. S. Marcus	P. A. Wieselmann
H. L. Graham	R. E. McPhail	Y. D. Willems
A. K. Griffith	J. Moses	R. D. Wright
J. J. Grimes	J. L. Openshaw	

Other Graduate Students

W. L. Bass
B. H. Bloom
A. J. Bodhe
J. R. Brown
S. Brueck
P. B. Clark
J. A. Davis
R. Domercq
D. L. Flamm
Z. Fluhr
J. C. Free
G. Goddard
R. S. Green
T. L. Hawk
W. Inglis
D. A. Kendrick
A. N. Kramer
N. Leal-Cantu
B. K. Levitt
M. Manove
R. L. Martin
T. E. McElroy, Jr.
P. F. Meyfarth
N. P. Negroponte
R. Parker
D. Prerau
W. D. Reinhardt
A. Scholl
T. W. Schwenke
C. L. Seitz
A. A. Scholl
S. Simpson
T. P. Skinner
E. Soltero
M. J. Spahn
B. T'sou
D. Udin
R. N. Wallace
D. A. Wright
I. T. Young

Undergraduates

E. L. Anderson
D. E. Avrin
A. G. Baisley
M. A. Becker
P. Bos
R. H. Bryan
F. H. Ciaramaglia
J. L. Fields
M. Fineman
W. J. Flor
A. S. Gevins
J. E. Gips
A. C. Goldstein
A. J. Gottlieb
R. P. Greer
D. M. Griffel
J. M. Grochow
K. Grossen
M. M. Hammer
T. M. Higgins
R. Hodges
L. C. Kaufman
D. W. Lee
D. W. Levine
L. K. Marquardt
W. H. McCandless
J. T. McGettigan
G. H. H. Mitchell
P. V. Mockapetris
L. H. Morton
A. Moulton
G. A. Moulton
W. M. Parks
A. Pawlikowski
K. T. Pogran
H. A. Radzikowski

Undergraduates (continued)

D. Rodman	M. Speciner	P. Vale
G. S. Rudgerman	J. R. Stone	F. N. Webb
H. L. Seldon	J. S. Sukonick	J. C. Yochelson
C. R. Smith	G. J. Sussman	B. L. Zimmerman
S. W. Smoliar	J. C. Thompson	

Technical Assistants

T. F. Callahan	R. Fidler	G. L. Rosen
D. E. Eastlake	J. S. Roe	

Operating Staff

G. Andrews	T. Dattilo	R. Moore
D. Borsini	R. A. Degan	G. Noseworthy
C. Brandon	D. Lang	M. Pagliarulo
K. Carley	J. McGillivary	D. Peaslee
J. Cohan	R. McNamara	E. Reardon
R. Crowley	P. Monaco	J. W. Waclawski

Technicians

P. J. Pinel	R. G. Splaine	N. F. Stone
R. J. Porcaro		

Administrative and Supporting Staff

C. S. Alles	C. D. Graceffa	J. A. Robinson
J. C. Anderson	D. L. Jones	E. M. Roderick
M. E. Baker	D. Kontrimus	D. C. Scanlon
T. Bergeron	T. Liotta	M. Stallings
A. W. Bowen	R. Long	S. L. Stone
J. M. Constantine	E. T. Moore	K. M. Tomaselle
B. L. Cullen	M. A. Nelson	C. M. White
K. F. Diamond	J. E. Piñella	
L. S. Ely	C. L. Robinson	

PREFACE

Project MAC was organized at the Massachusetts Institute of Technology in the spring of 1963 for the purpose of conducting a research and development program on Machine-Aided Cognition and Multiple-Access Computer systems. It operates under contract with the Office of Naval Research, acting on behalf of the Advanced Research Projects Agency of the Department of Defense.

The broad goal of Project MAC is the experimental investigation of new ways in which on-line use of computers can aid people in their individual intellectual work; whether research, engineering design, management, or education. One envisions an intimate collaboration between man and computer system in the form of a real-time dialogue where both parties contribute their best capabilities. Thus, an essential part of the research effort is the evolutionary development of a large, multiple-access computer system that is easily and independently accessible to a large number of people, and truly responsive to their individual needs. The MAC computer system is a first step in this direction and is the result of research initiated several years ago at the M.I.T. Computation Center.

Project MAC was organized in the form of an interdepartmental, interlaboratory "project" to encourage widespread participation from the M.I.T. community. Such widespread participation is essential to the broad, long-term project goals for three main reasons: exploring the usefulness of on-line use of computers in a variety of fields, providing a realistic community of users for evaluating the operation of the MAC computer system, and encouraging the development of new programming and other computer techniques in an effort to meet specific needs.

Faculty, research staff, and students from fourteen academic departments and four interdepartmental research laboratories are participating in Project MAC. For reporting purposes, they are divided into sixteen groups, whose names correspond in many cases to those of M.I.T. schools, departments and research laboratories. Some of the groups deal with research topics that fall under the heading of computer sciences; others with research topics which, while contributing in a substantive way to the goals of Project MAC, are primarily motivated by objectives outside the computer field.

The purpose of this Progress Report is to outline the broad spectrum of research being carried out as part of Project MAC. Internal memoranda of Project MAC are listed in Appendix A, and MAC-related theses are listed in Appendix B. Some of the research is cosponsored by other governmental and private agencies, and its results are described in journal articles and reports emanating from the various M.I.T. departments and laboratories participating in Project MAC. Such publications are listed in Appendix C of the report. Project MAC Technical Reports are listed in Appendix D.

Robert M. Fano

Cambridge, Massachusetts

ADMINISTRATION

CTSS Maintenance, Administration, and Operation

Academic Staff

R. M. Fano - Director

Non-Academic Research Staff

R. G. Mills - Assistant Director

Administrative Staff

M. L. Gabral

J. A. Gunn

A. J. Saltalamacchia

T. H. Van Vleck

<u>CTSS Maintenance, Administration, and Operation</u> - Richard G. Mills, Malcolm M. Jones, and Thomas H. Van Vleck

During the year 1966-67, the CTSS system was treated more and more as a research tool, rather than as an object of research. Many minor bugs, inconveniences, and rough spots were corrected, but no major change was made to the system.

Over 3000 hours of 7094 time were charged to the user community, which numbered about 350. The average user of the system had about 35 files totalling 160 records, and 30 links in his file directory — these averages remained quite constant through the year.

BLANK PAGE

ARTIFICIAL INTELLIGENCE

Academic Staff

M. Blum
M. L. Minsky
S. A. Papert

Non-Academic Research Staff

R. J. Abbott
G. W. Baylor
M. D. Beeler
H. K. Beller
W. Bennett
T. O. Binford
H. E. Brammer
M. E. Denofsky
R. Greenblatt
R. W. Gosper
W. H. Henneman
P. Himot
J. T. Holloway
M. J. Lennon
W. R. Noftsker
J. S. Roe
P. Samson
D. Sordillo
G. Spielman
G. Voyat
J. L. White

Research Assistant and other Students

R. Baecker
A. G. Baisley
M. A. Becker
W. T. Beyer
B. H. Bloom
R. J. Bobrow
E. Charniak
S. D. Crocker
S. Geffner
A. C. Goldstein
A. K. Griffith
A. Guzmán
B. Harvey
T. L. Jones
L. Kaufman
L. J. Krakauer
D. W. Lee
M. E. Manove
W. A. Martin
W. H. McCandless
G. H. Mitchell
J. Moses
G. A. Moulton
W. M. Parks
D. N. Perkins
D. Prerau
D. Rodman
S. Simpson
C. R. Smith
S. W. Smoliar
M. Speciner
J. S. Sukonick
G. J. Sussman
J. C. Thompson
D. L. Waltz

Guests

A. Forte - Yale University
R. Silver - MITRE Corporation

Research on Intelligent Automata* - Marvin L. Minsky and Seymour A. Papert

The major goal of this project is to develop better sensory and motor devices, together with controlling programs, for computer-controlled manipulation. Our current target is to assemble enough facilities so we may give a machine a goal-type task statement to act upon.

To achieve this, the programs have to do a number of things: analyze the goal statement; analyze the visual scenes mentioned in the statement, using visual equipment; form a plan of action that will achieve the goal; control motor organs, using visual and tactile feedback to perform the planned actions; and monitor the whole action sequence, predicting its course and checking expectations, with continuous capability for revising the action plan should difficulties arise.

Earlier reports proposed how some of this might be done. As might be expected, the visual and analytic problems outweigh the purely mechanical ones in general difficulty. Our general approach is to regard the problem of vision as a broad research area that needs specific research on both its theoretical and practical aspects. Since so little is really known about the practical aspects, our policy has been to try to make sure that we keep open at least two possible ways to complete each path.

A. THE VISION LABORATORY

Our goal is to build up a set of modular programs as equipment for a "vision laboratory". Most of these programs carry out operations considered to be likely components of many different, experimental vision programs. Others are of a utility nature: display programs, programs to store and retrieve visual information, etc.

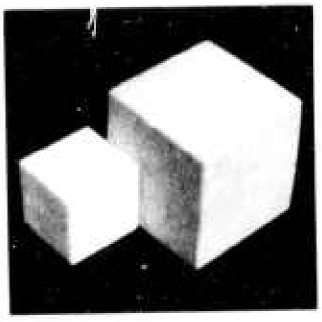
1. Analysis of a Simple Scene

Scene-analysis is the result of interaction between optical data coming from the eye, and knowledge about the visual world stored in the programs and their abstract data files. The interaction must be bilateral in that each helps to select what is done with the other. It is impractical "blindly" to apply sophisticated computations to all of the millions of resolvable points in a picture and it is even more impossible "blindly" to match all parts of the picture to all possible object configurations.

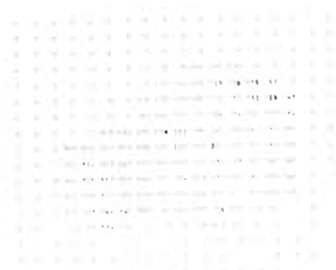
---

*Abstracted from Status Report II: Research on Intelligent Automata, Project MAC, 1967

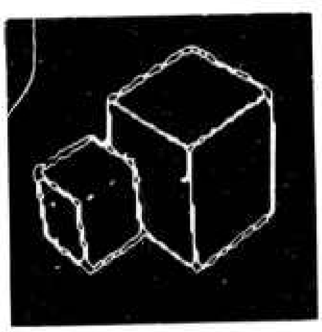
Consider a simple scene containing two white blocks on a black table. We use rectilinear objects because there are many still-unsolved problems in dealing with curved surfaces.



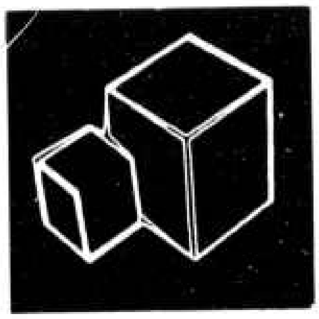
A coarse scan over this scene yields the following intensity distribution: the numbers are logarithms of the light intensity — to make the programming more independent of changes in general light level.



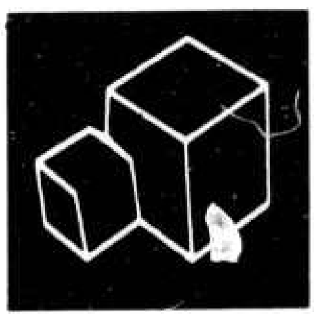
The cost in computer time is small to make a coarse scan like this, and yields a lot of information to start the analysis. A program now picks out "homogeneous squares" — those squares in the coarse raster whose corners have nearly the same values —



and another program, TOPOLOGIST, groups them into connected regions and finds the boundaries of these regions.

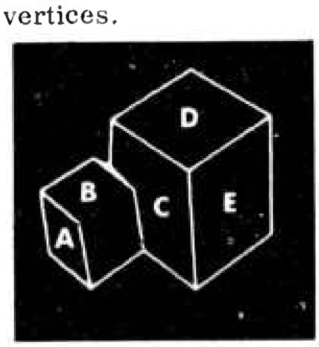


Next, a program attempts to extend each of these, using progressively finer measurements so that one obtains a sharper idea of the region's boundaries and we have some of the effect of a very fine scan

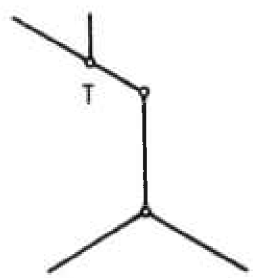


without having to look at the whole scene with full resolution. In effect the coarse scan is used to decide which areas need finer analysis.

Now a new program, POLYSEG, tries to fit the boundaries to straight lines, and it does indeed find good fits in this case. The resulting analysis gives us the first chance to start working abstractly instead of continuing in "picture-point space". The line analysis yields a LISP expression (see MAC-M-134) which can be interpreted by a display program as a diagram with vertices.

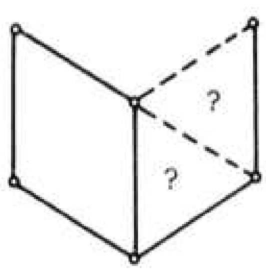


From this description we can take two paths. One path can already "guess" that these are two rectangular objects, using the general principle that "T-joints" are usually optical rather than structural



phenomena, and by using built-in knowledge that sometimes such an object can look edgeless when the light on the two faces is about equally bright, for the line dividing them may be invisible or at least hard to see.

Another path the program can take is to classify the two-dimensional shapes first: A, D, and E are recognized as close to parallelograms; C is recognized as "probably" a parallelogram (because it could be one if "continued" through its one T-joint), but region B is "non-standard". Using certain general principles, the program proposes alternate corner connections that might be there and perhaps were



missed in the coarse measurements. An especially sensitive "line-verifying" program is applied, and often it will pick up a boundary that

was missed in the less-intelligent preliminary region-survey phase. Here it can be practical to apply a very stringent and sensitive test, because the program knows very accurately where the line should be, if it really exists at all. For example, even if the two faces have almost equal illumination the eye could pick up a thin, faint highlight from the edge of the cube. It would have been hopelessly expensive to look for such detailed phenomena over the whole picture at the start.

But efficiency, i.e., amount of computer-time, is not the only important issue. The real problem is how to analyze the scene at all. For the problem of taking a two-dimensional image (or several such images), and constructing from it a three-dimensional interpretation, involves many things that have never been studied, to say nothing of being realized on a computer.

2. Problems in Analyzing a Visual Scene

Among the facilities that must be available are:

a. Spatial frame-of-reference: setting up a model of the relation between the eye(s) and the general framework of the physical task, i.e. where are the background, the "table" or working surface, and the mechanical hand(s)?

b. Finding visual objects, and localizing them in space with respect to the eye-table-background-hand model.

c. Recognizing or describing the objects seen, regardless of their position, accounting for partly-hidden objects, recognizing objects already "known" by descriptions in memory, and representing the three-dimensional forms of new objects.

d. Building an internal "structural model" of what has been seen, for the purpose of task-goal analysis.

These are all difficult problems, both theoretically and practically, because the visual appearance of a scene of objects depends drastically on so many factors, none of which can be neglected! Among these are: both the camera's focus and its depth-of-focus, illumination of the objects, perspective aspect and distance effects, accidental vs. essential visual features, and "temporal" visual features.

3. "Philosophy" of the Vision Laboratory

Our approach is based on the central principle that the vision problem is a very complicated one. A visual scene can be analyzed only by using a great deal of information about the physical world. The system must know about real objects, and how visual phenomena can affect their

appearances; it must know about mechanical stability, rigidity, support; it must understand perspective, illumination, color and shadows. To solve harder problems, the program must permit interactions of stored knowledge at all levels.

There are many simple techniques that are very useful in a vision system, some of which will be listed in the following sections. The point is that each of the simple methods works on some problems, but not on all. One could take each and try to extend it to be more universal. But our experience is that cost and complexity shoot up rapidly when a method is pushed even a little past its "natural" limits. Our counter-approach is to absorb all the methods we can, filing them with statements about where they can help, and in what situations they are liable to fail. Then we try to put the major effort into organizing higher-level programs to use these growing banks of knowledge.

4. Techniques for Localization of Objects in Space

Localization of an object includes finding its spatial orientation as well as its distance. A richer family of methods, for localizing objects whose shapes are known, depends on analyzing the two-dimensional shape as a perspective projection. Roberts* showed how, for reasonably large plane-bounded objects, one could use this kind of analysis to build up a space model, and he extended the analysis to handle combinations of rectilinear prismatic objects — thus pointing out that with an auxiliary three-dimensional shape-determining problem solver, one could escape, to some extent, the requirement that the object's shape be known in advance.

In the current hand-eye system, the "calibration" phase — measuring the relations between the hand and the eye — is done using this principle. In one system, the mechanical arm traces out in space the outlines of a large object of known shape. The eye measures the apparent (visual) location of its vertices. Then, using its knowledge of where they are in "arm-space", the program calculates from the perspective appearance of the "object" where the eye must be located. In a second system, still under development, an actual object is placed in view; its relation to the eye is determined, and its relation to the hand also determined by gross visual measurements and finally checked by tactile information. (All the calculations are done using homogeneous coordinate vector programs, following the suggestion of Roberts.)

---

*Roberts, L.G., Machine Perception of Three-Dimensional Solids, Technical Report No. 315, Lincoln Laboratory, M.I.T., May 1963, p. 146

5. Acquisition of Contours and Regions

a. Boundaries

We have been engaged in extensive work on edge-tracing. A tracking program was developed, chiefly by R. Greenblatt and J. Holloway, and described in A.I. Memo 101. It has been expanded to record edge-vertex relations and to use all incident lines to estimate vertex locations. The program makes the hypothesis that it is seeing a polygonal prism resting upon a known surface. It identifies vertical edges and upper and lower surface edges, using a logical minimum of the edges produced by the edge-tracker, and uses the others to improve its estimate of where these planes are. Using this system, Gosper has developed a program that can find, pick up, and sort by size, small rectangular objects. It works on objects down to about one twentieth of the field of view of our old eye, TVB, whose effective resolution is only about 500 points, picking up one-inch cubes with errors of about one eighth of an inch in all dimensions — without stereo.

b. Region-oriented methods

A system whose initial contact with the world is through a system of line-following, knows objects or regions as constructs from their bounding lines. Region-oriented systems acquire regions as their primary first contact with the world and later find boundaries of regions. The TOPOLOGIST system illustrates this approach.

The program POLYSEG takes a list of points and tries to fit a polygon to them. Its output is the set of vertices of the proposed polygon. It decides, for example, using various parameters (which should be set more and more rationally as we progress), whether two apparent lines making an angle of 175° should be regarded as one line or two, whether a very small apparent side should be eliminated as probably due to noise or resolution-limit error, etc. When such a process is finished, we have a set of "geometric regions" described in terms of abstract lines and curves. The next problem is to decide how "adjacent" regions are related: when two edges found by different methods are really the same, and when several regions really share a vertex.

At this point, with the scene tentatively mapped out into reasonably well-defined geometric regions, we can proceed to attempt to find the objects.

6. Discovering the Objects in a Scene

Processes of the kind discussed in 4. and 5. yield a "symbolic picture" as a collection of two-dimensional geometric entities — points, lines, regions with properties such as position, length, brightness,

etc. -- and relations such as in, next to, etc. The next problem is to postulate a configuration of objects in three-dimensional space which could give rise to the picture.

"Object analysis" and "object recognition" are the two processes involved. Object analysis corresponds to the process celebrated in psychological literature as "figure-ground discrimination" which is supposed to pick out those parts of a visual tableau to be seen as objects (figures) rather than background. The problem of object analysis is to try to decide which regions come from the same physical three-dimensional objects, and which come from different objects. Once enough information is available, one can go on to try to identify the objects' forms and locations in space.

7. Recognition of Objects in Three Dimensions

Although the literature contains a good deal of material on recognition of two-dimensional forms, e.g., character-recognition methods, this literature is of little help for the problem of recognizing three-dimensional objects by their projections. This is because the problems are so completely different: the two-dimensional problem is essentially to normalize the object against the effects of size and position variations — once this is done the figures can usually be recognized by one or another variant of matching or correlation. In the three-dimensional problem, we have to face not only this difficulty, but also the more critical problems of perspective distortion, self-occlusion, occlusion by other bodies, and all the optical variables due to shadows and lighting. Mathematically, the difference is very sharp; in the two-dimensional case the appearances of the same object are related by simple one-to-one transformations; in the three-dimensional case a single appearance simply doesn't contain enough information to determine the others. Recognition is accomplished only if the machine contains enough information to recognize any view of the object. Guzmán's report, MAC-TR-37, gives a detailed description.

8. A More "Vertically Organized" Vision System

In the foregoing sections we have described a sequence of techniques for picture analysis: finding "homogeneous" regions; then finding boundaries; then object analysis; and finally object recognition. This general concept of proceeding through such a sequence of phases leads to a very simple, easy to understand and easy to debug, scene-analysis system. But it is also very inflexible and makes it difficult to achieve more ambitious goals involving visual ambiguities, or even modest goals under poor visual conditions, i.e., with objects whose surfaces are not particularly homogeneous to begin with. What is needed, as we stated at the beginning, is to develop a less stratified

system — a more "vertical" one — in which the knowledge at different levels can interact. This is prevented by the "horizontal" organization in which all processes on each level are completed before going on to the next level. Required is a system with less stratification into independent successive phases.

9. Data Structures

The concept of the vision laboratory in general, and the vertical vision system in particular, depends heavily on finding a sound approach to communication between programs. So long as the programs were relatively few and simple, it was sufficient to write each one as, say, a LISP function and to be sure that all users knew the proper format for the function's arguments and values. Thus the horizontal vision system could find the shape of the region containing the point P by being asked to compute the value of

(SHAPES (POLYSEG (BOUNDARY (REGION P))))

and LISP would return the answer (PARALLELOGRAM) if that were the correct answer. For, the value of (REGION P) is a list of the points of the region containing P, (BOUNDARY L) gives the list of lists of L's boundary (one list for each part of the boundary), and this is the input form for POLYSEG, etc.

For the vertical system this simple scheme can't be used, for the problems are too complicated for any such fixed form of data-structure. As the system operates it discovers new facts about the scene at different levels; as the shapes are analyzed, lines and vertices may be reassigned to different temporary assignments. Eventually, we will have to develop a standard way of keeping track of all these things, in a way that will allow reasonably efficient information-retrieval by higher-level deductive programs. At present we are using a simple, general, but very inefficient system in which linkages between elements are stored as entries on their property-lists. It is hoped that, as it becomes clearer which operations on this data are most used, we will be able to sharpen up our ideas about how better to structure the program's knowledge about any scene.

10. Research on Human Vision

It would be fine if, in trying to develop methods for vision in machines, we could directly apply knowledge about how vision is achieved in animals, and particularly in man. The problem here is not so much in applying knowledge, but in discovering it, for very little is really known about how we see. There is some knowledge as to how the retina works, but this concerns chiefly the simplest kinds of "preprocessing". We know ways to sharpen contours, emphasize gradients,

brighten colors, etc, but much more remains to be done in those areas. Much less is known about how whole objects are perceived in humans. Because of our combination of interests and facilities we have maintained a small activity in the study of human visual perception. One interesting question pertains to what strategies are used by people in searching through a scene to find a bit of information they have been assured is in the scene. This was the subject of a doctoral thesis by Henry Beller, who did experimental work in our laboratory while a graduate student at Brandeis. Another experimental thesis by Arnold Stoper (also of Brandeis) was a study of the nature of human perception during eye motion. A third area of interest is the nature of perceptual judgments of motion, which was the subject of the doctoral thesis of Gilbert Voyat, who did this work in Piaget's Institute in Geneva and is now on our staff.

In the coming year, Stoper hopes to realize one of our long-term goals: finding enough about the role of human eye motions in scene-analysis to suggest at least a hint about visual strategies. Our hopes are not extremely high in this area, for evidence has been accumulating that people can solve some pretty complicated visual problems without a very great amount of eye-motion. (This is not too surprising, for one can make at most a few motions per second, and can orient oneself in a scene in a very few seconds.) Nonetheless, we would like to know what role this activity has. As the vision laboratory develops, we expect to find increasing interest among allied workers in using its facilities to simulate and evaluate theories of human vision.

B. HARDWARE

1. Optical Hardware

A new input camera, "TVC", is used as a visual input device. The chief innovation in TVC is program-controlled signal-to-noise ratio: the programmer selects the degree of brightness resolution required and the camera automatically controls the exposure to collect the appropriate amount of information. A second innovation that substantially improves performance in certain situations is programmable "dark cutoff": the video processor makes a preliminary estimate of the brightness based on the rate of detection of the first few photons, and if this statistical sample falls below a given threshold, the measurement is not made. TVC is essentially operational, although there remain some problems with its deflection hardware. Its spatial resolution is of the order of 1000 lines, and its intensity resolution is better than one part in 64 over a 64-to-one dynamic range. We expect TVC to be an excellent research tool, suitable not only for real-world scene analysis, but also, because of its high measurement accuracy, to be

usable for some types of analysis that require moderate photometric precision.

A computer-controlled optical system has been constructed and is being installed on the vidissector-camera TVC. This makes it possible for a vision program to select: 1) focus of lens, 2) iris diameter (hence depth of field), 3) color filters, and 4) choice of two stereo views.

It was necessary to build this because, although there exist remote-controlled optical units in the television industry, none of the commercial units have the speed or flexibility appropriate to computer vision. When we have accumulated enough experience with this unit, over the next few months, we plan to design and build a "final" version which, like TVC, can be made easily available to other groups who want to work in this area. The current unit was designed chiefly by T. Callahan.

2. Arms and Hands

The modified AMF Versatran manipulator, MA-2, has continued to serve with minor changes in control hardware. The hand is being rebuilt to provide better force-feedback facilities. The new arm, MA-3, described in detail in our Status Report I, is being modified for greater strength, speed and mobility. At present, its capacities are roughly comparable to the human arm in the following aspects: it is about as mobile, perhaps twice as strong, and half as fast. We plan to revise it after accumulating more experience with it: current plans for the "final" version are pointed toward obtaining a simpler external appearance with no externally vulnerable parts, and provision for force-reflecting servomechanisms.

3. Computers

It appears that at some time within the next year, it will be desirable to add a second processor to our central computer, the PDP-6. The initial selection of the PDP-6 computer was based in part on the fact that it was the only major-machine available that had built-in room for expansion, not only in input-output channels (which have grown steadily) but also in memory channels and in additional central-processor capacity. The manufacturer has continued to develop this concept and can offer a completely program-compatible second processor. This new processor, called the PDP-10, is about twice as fast as the present PDP-6, and if added with a small amount of its own very fast memory ought to triple the system's bulk computation rate.

4. Laboratory Area

The laboratory has been re-designed to facilitate maintenance, modifications, and debugging of the on-line machinery. The main requirements were to make available clear visual spaces and good lighting, but still have access to consoles, electrical and hydraulic controls, and power lines. The chief improvement is a raised floor under which is a network of these lines, with junction boxes for input-output control boxes that can be hand-held while working on the equipment. A design is partly completed for a hand-held portable computer console.

Systems and Time-Sharing* - Marvin L. Minsky and Seymour A. Papert

A. THE PDP-6 TIME-SHARING SYSTEM

A time-sharing system for the PDP-6 went into operation in July 1967, and currently provides service to the eight consoles which are connected to the PDP-6.

Our PDP-6 is equipped with a Fabritek memory of 256K directly addressable words with a 2.75 microsecond cycle time. The memory is utilized to avoid swapping user programs and allows large dynamic I/O buffering. The overhead is therefore small enough to allow 1/30 second quantum per procedure. Such a small quantum essentially guarantees immediate response for interaction-limited programs. (In CTSS, response time may be longer than 60 seconds.)

The system has the ability to be multi-programmed. Any procedure may create and closely control inferior procedures, all of which will share time. Inferior procedures may generate interrupts to the superior procedure which created them, and superior procedures have the ability to start, stop, modify, and destroy inferior procedures. It is also possible for two arbitrary procedures (even two not started by the same user) to treat each other in the manner of I/O devices and to communicate directly. Also, inferior procedures may be "disowned" so that they operate as independent jobs. Disowned jobs may logout by themselves or may be reattached to a user (not necessarily the originator).

Most of the code in the time-sharing executive is re-entrant, so that procedures may be interrupted while a request to the executive is

---

*Abstracted from Status Report II: Research on Intelligent Automata, Project MAC, 1967

being processed. The longest time in which interrupts are disabled is about 200 microseconds, and this happens infrequently. Capability for real-time response will be added, although service to devices like the arms is rapid enough now without special scheduling.

Most time-sharing systems interface with the user at two levels. One is the command level which interprets commands from the user's console and the other is at the machine-language level which interprets executive calls made by a user's program. In our time-sharing system, only the latter level exists, with the sole exception that an interrupt may always be caused from the user's console. When a user notifies the system of his presence at a console, he is presented with a program which will interpret commands to start up or stop inferior procedures, but the user is free to replace this top-level program with one of his own. One consequence of this arrangement is that the user's console is treated exactly like any other I/O device and it may be passed around from procedure to procedure under program control. (The sole exception, of course, is the ability to cause interrupts to a superior procedure through the console.)

Input and output are device-transparent and a superior procedure may reassign the device for its inferior procedures.

The scheduling algorithm attempts to equalize response to consoles, and also to equalize the time devoted to each job of a given user.

B. THE PDP-6 LISP PROGRAMMING SYSTEM

The higher-level language used for most of the vision laboratory program is the PDP-6 LISP System. This system is based chiefly on the LISP 1.5 programming language, but has been extensively modified in a number of ways. These include many new functions and services, including facilities for linking with programs written in other languages. The most important extensions of the language include:

1. On-line facilities for use in and out of time-sharing,
2. Real-time control of most input-output devices by internal LISP functions,
3. File and retrieval functions for all storage devices, and
4. Special facilities for picture-arrays and real-time use of the eyes.

Also, a major extension of the language, the CONVERT string-matching manipulation language, is operating as a LISP-embedded language, with its own CONVERT-to-LISP compiler. The PLANNER deductive system developed by Carl Hewitt is also being embedded, for use with the proposed "vertical" vision system.

Theoretical Work on Vision* - Marvin L. Minsky and Seymour A. Papert

We believe our theoretical results to be of considerable importance both in the domain of pattern-recognition and for computer science in general. A number of links have developed, strengthening both our practical technology and some theoretical aspects of Automata Theory, Threshold Function Geometry, and Learning Theory.

A. LINEAR SEPARATION THEORY

Much is now understood about the "perceptron" type of pattern-recognition device, both in the nature of the patterns it can recognize and in the behaviour of its learning mechanism. Previously, some properties of pictures, like "connectivity", were fundamentally beyond the reach of perceptrons, but some others, like "convexity", were recognizable. Since then we have results in a number of new directions. One basic result is that even if a class of patterns can be recognized by themselves, the perceptron will, in general, break down when the patterns are placed in a background context. Another result shows that topological properties are unrecognizable in general, with one peculiar exception: the "Euler characteristic" function. Another set of results appears to clarify the connection between "normalization" — a practical technique usually used in pattern-recognition systems — and the structure of perceptron-like machines. Finally, the character of the "learning" process for such machines has been interpreted in a better geometric representation, leading to better estimates of how efficient the process can be. The results will appear in book form entitled Perceptrons, M.I.T. Press, in 1968.

B. THEORY OF PATTERN RECOGNITION BY TWO-DIMENSIONAL FINITE AUTOMATA

This theory, developed by Manuel Blum and Carl Hewitt, also has as its goal the development of a mathematical theory of pattern recognition. They have extended classical automata theory, which deals with one-dimensional "tapes", to a theory dealing with two-dimensional tapes in order to study various properties of patterns, e.g., connectedness, convexity, translation of regions, etc., and have devised means whereby an automaton may detect these properties. The theory has three aspects:

---

*Abstracted from Status Report II, Research on Intelligent Automata, Project MAC, 1967

1. The study of how various properties of patterns are altered when the patterns are mapped from Euclidean space onto a checkerboard grid;

2. The design of automata that detect certain properties of two-dimensional patterns, e.g., whether the pattern is symmetrical or not; and

3. The demonstration that certain properties, though well-defined and easy to recognize by serial-computer programs, cannot be detected by a classical finite-state automaton.

We feel that these results of Blum and Hewitt are important in that they give us directions to search for sound theoretical ideas about classifying patterns — important for addressing the difficulty of new pattern-analysis problems. Perhaps of more general significance to "computer scientists" is that they have established a link between the very active mathematical fields of activity called "automata" and "mathematical linguistics" and some practical physical-type problems. This is important in guiding a new science toward profitable areas of application.

A Primitive Recognizer of Figures in a Scene — Adolfo Guzmán

DT is the name of a program, written in CONVERT* which partially achieves the problem of analyzing a given scene, as for instance from a TV camera or a picture, in order to recognize, differentiate, and identify desired objects or classes of objects (i.e., patterns) in it. Two inputs to the program determine its behavior and response:

1. The scene to be analyzed, which is entered in a symbolic format (it may contain 3-dimensional and curved objects);

2. A symbolic description — called the model — of the class of the objects we want to identify in the scene.

Given a set of models of the objects we want to locate, and a scene or picture, the program will identify in it all those objects or figures which are similar to one of the models, provided they appear complete in the picture (i.e., no partial occlusion or hidden parts). Recognition is independent of position, orientation, size, etc.; it strongly depends on the topology of the model.

---

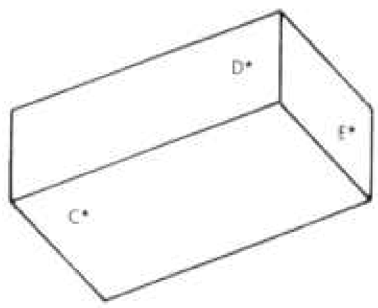
*Guzmán, A. and H. V. McIntosh, "CONVERT", Communications of the ACM, vol. 9, no. 8, August 1966, pp. 604-615

Important restrictions and suppositions are: a) the input is assumed perfect - noiseless - and highly organized; b) more than one model is, in general, required for the description of one object; and c) only objects which appear unobstructed are recognized. Work is continuing in order to drop restriction c) and to improve a).

A 1967 E. E. Masters thesis (see Guzmán, Appendix B) contains fuller description of this; an internal memorandum (see MAC-M-342, Appendix A) contains details of the program.

An example of scene analysis follows. The models CUBE (Figure 1) and HOLLOWBRICK (Figure 2) are used to analyze the scene in Figure 3. The scene in Figure 3 is analyzed by DT, the program, in the PDP-6 computer. The symbol # marks the lines typed by the user.

```
#  CONV 4                                             Bring the CONVERT proces-
                                                      sor from tape 4.

#  (UREAD DT LISP 5 ↑Q ↑W)                            Load the file containing DT,
                                                      the recognizer.

#  (UREAD EX2 LISP ↑Q ↑W)                             Bring the scene EX2 into
                                                      memory. (see EX2 Figure.)

#  (UREAD MOD2 LISP ↑Q ↑W) (IOC V)                    Load the models.
   (V)

#  (DT (QUOTE CUBE) (QUOTE EX2)                       Look for CUBEs in EX2.
   (CUBE 1. IS (A B C)                                (See EX2 Figure.)
   (CUBE 2. IS (J L M))                               2 cubes are found.
   (D E F G H I K N O P Q R S T U V W X Y Z)          Remaining of scene.

#  (DT (QUOTE CYLINDER) (QUOTE EX2))                  Look for cylinders.

   (CYLINDER 1. IS (E D))
   (CYLINDER 2. IS (G F))
   (A B C H I J K L M N O P Q R S T U V W X Y Z)      ← remaining of scene.

#  (DT (QUOTE HOLLOWCYLINDER) (QUOTE EX2))
   (HOLLOWCYLINDER 1. IS ( T U S))
   (A B C D E F G H I J K L M N O P Q R V W X Y Z)

#  (DT (QUOTE HOLLOWBRICK) (QUOTE EX2))               ← see HOLLOWBRICK Fig.
   (HOLLOWBRICK 1. IS (N O P Q R))
   (A B C D E F G H I J K L M S T U V W X Y Z)
```




```
(DEFPROP (CUBE)
((C* (NEIGHBOR E*)
   (NEIGHBOR D*)
   (SHAPE PARALLELOGRAM))
 (D* (NEIGHBOR C*)
   (NEIGHBOR E*)
   (SHAPE PARALLELOGRAM))
 (E* (NEIGHBOR D*)
   (NEIGHBOR C*) (SHAPE PARALLELOGRAM) ) )
   REGIONS
```

Figure 1. CUBE — A Model. (It is really a parallelepiped.)

```
(DEFPROP HOLLOWBRICK
  ( (D* (NEIGHBOR E*) (NEIGHBOR F*) (NEIGHBOR G*) (NEIGHBOR H*)
        (SHAPE (PARALLELOGRAM INSIDE PARALLELOGRAM)) )
    (E* (NEIGHBOR U*) (NEIGHBOR F*) (SHAPE TRAPEZ) )
    (F* (NEIGHBOR E*) (NEIGHBOR D*) (SHAPE TRAPEZ) )
    (G* (NEIGHBOR D*) (NEIGHBOR H*) (SHAPE PARALLELOGRAM) )
    (H* (NEIGHBOR D*) (NEIGHBOR G*) (SHAPE PARALLELOGRAM) )
REGIONS
```

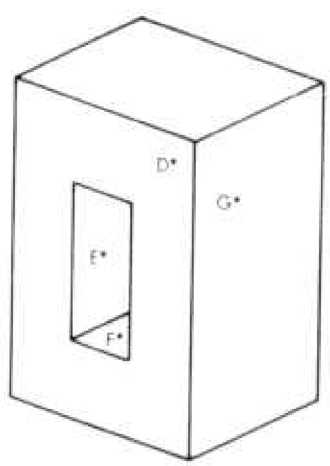


Figure 2. HOLLOWBRICK — A Model.

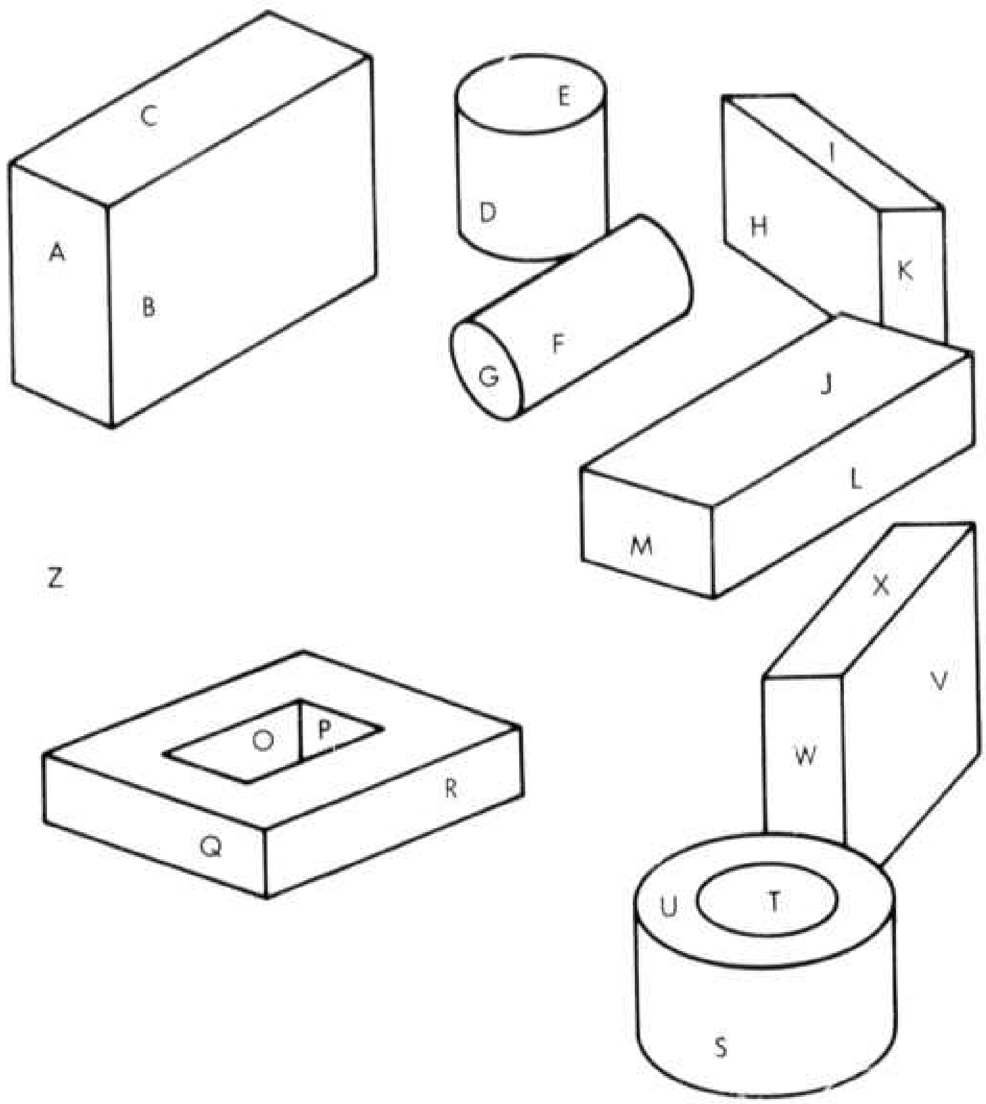


Figure 3. EX2 — A Scene.

A Miscellany of CONVERT Programming - Harold V. McIntosh and Adolfo Guzmán

CONVERT shares with other programming languages the facility to evaluate the language and to learn its use if it is possible to scrutinize a representative sample of programs which effects typical but simple and easily understood calculations. Consequently we have assembled several examples of varying degrees of difficulty in order to show CONVERT in action. In each case the CONVERT program, written as a LISP function ready for execution in CTSS, is shown, together with the results of its application to a small variety of arguments, and a general explanation of the program, its intent, form of its arguments and method of its operation. When the notation CLOCK (()) . . . CLOCK (T) appears, the time of execution has been determined, and is shown, in tenths of seconds immediately after print-out. (See MAC-M-346, Appendix A, for a detailed description of this work.)

Representation of Geometric Objects by Circular List Structures - David N. Perkins

In the search for successful means of machine simulation of visual processes, there has always been the problem of representing objects in a symbology geometrically accurate, natural to man's intuitive visual sense, and as convenient as possible for a computer to store and process. These three sides of the problem are naturally in some competition.

A weakness in current schemes for representing objects is inability to represent with any accuracy the richly structured objects that one finds in the real world, such as a telephone with curved surfaces, many highlights, complicated dial, printed letters, and such. Therefore I sought methods of representing the shape of an object accurately (leaving aside questions of color, highlights, etc., for awhile).

My conclusion was that shapes could be captured in symbols by a latticework of "nodes" connected by "links". Nodes are symbolic expressions representing local behavior at a point on a surface. Nodes are composed of spikes which are expressions for behavior in a certain direction from a certain point. The node language can represent simple curvature, more complicated intersections of planes such as at the corner of a cube, and many further intricacies of local behavior.

An object usually consists of repetition of a few kinds of local structure. For instance, a cube basically has only three kinds of local behavior - plane, 90 degree edge, and corner. One chooses the nodes to represent the places of high information content in the object, such as corners and edges.

A link is basically a pair of spikes from different nodes, plus information about the distance between the nodes. When two spikes are linked then the behavior of the surface between the two spikes is redundant, and represented by either spike.

Since any node might have spikes linked to numerous other nodes, they should be represented by a latticework of cells inside the computer, which would form an analog of the topology of the original object. This required circular list structures and therefore promoted the development of functions written in the LISP language to deal with such structures. These include a pattern-recognition program for dealing with arbitrary and possibly circular lists. The patterns may themselves be circular.

Vision During Pursuit Movement: The Role Of Oculomotor Information - Arnold Stoper

When exploring a stationary visual scene one makes rapid "saccadic" eye movements from one fixation point to another. For some reason the image displacement over the retina caused by the saccadic eye movement does not result in apparent motion of the visual scene. A closely related problem is the fact that the smooth motion of an illuminated object in an otherwise dark room is seen even though the eye follows the object so as to keep its image stationary on the retina. (See Stoper, Appendix B.)

The generally accepted explanation for both these phenomena is some version of the "cancellation theory," originally suggested by Helmholtz. This theory assumes that a centrally produced oculomotor signal accompanies the command to move the eye. This oculomotor signal is assumed to anticipate and cancel out the exact amount of image displacement which would be caused by the eye movement. If there is no image displacement, as in the case of an object pursued in the dark room, this same oculomotor signal would cause the perception of the object motion.

One version of this, called the "position cancellation" theory, assumes that the oculomotor signal operates to change the apparent direction of gaze so that it always corresponds to the actual direction of gaze. This process would result in veridical localization of objects regardless of the position of the eye. A stationary object would thus appear to be in the same location both before and after a saccade, and would therefore appear to be perceptually stable. The perceived motion of a pursued object is assumed to be due to the perception of its changed location at successive instants of time.

Another version — the "motion cancellation" theory — assumes that image displacement over the retina gives rise to a "motion signal". The oculomotor signal is assumed to consist of an equal but opposite

motion signal. During a saccade the two signals would cancel each other. When an object is pursued there would be no retinal motion signal. The oculomotor signal would thus remain uncancelled and would result in perceived motion of the object.

Since the time of Helmholtz much evidence has been gathered to indicate that the smooth "pursuit movement" of the eye is of a different nature, and is under different centers of control, than the saccadic movement. The major purpose of this research has been to show that these two eye movements are accompanied by different types of oculomotor signal, and that both versions of the cancellation theory are wrong in assuming that one and the same oculomotor signal is responsible for both perceived stability during the saccade and perceived motion during pursuit.

The speed of the saccadic eye movement is such that there is functional blindness during the time the eye is actually moving. Hence the apparent motion which must be "cancelled" to achieve perceptual stability is due to essentially stroboscopic stimulation; i.e., there is a stationary image appearing during the first fixation at one retinal location followed after a short time by the stationary image appearing at a different retinal location during the second fixation. The same pattern of excitation, if applied to a stationary retina, would result in apparent motion. Thus the oculomotor signal which accompanies the saccade somehow prevents stroboscopic stimulation from giving rise to the perception of motion. (This has been demonstrated under experimental conditions by Irwin Rock.) In the present experiments we investigated the influence of the oculomotor signal which accompanies pursuit movement on the perception of stroboscopically presented stimulation.

In condition 1 of experiment 1 the subject was instructed to follow a small point of light moving smoothly across his visual field at velocities ranging from 9°/sec to 27°/sec. The subject readily perceived this motion, and it was assumed that this perception took place by virtue of the oculomotor signal. A vertical line stimulus was flashed twice in the same physical place during each trial. The time interval between the flashes was varied. Since the eye had moved some distance during the time between the flashes, the stimuli excited two different retinal locations. The subject was asked to report any occurrence of stroboscopic motion. In the control condition, the subject was instructed to fixate on a stationary point, and it was assumed that no oculomotor signal was present. The retinal excitation pattern of condition 1 was duplicated. Since the image displacement in condition 1 was due entirely to the eye movement, the cancellation theory would predict that it would be entirely cancelled by the oculomotor signal.

It was found, however, that stroboscopic motion occurred frequently in condition 1 at certain time intervals. Further, no significant difference in the occurrence of stroboscopic motion was found between the two conditions. This indicates that the oculomotor signal had no tendency to cancel this motion.

In experiment II a similar stimulus presentation was used but the subject was asked to report on the apparent relative location of the two stimuli. As in experiment I, the movement of the eye during the time between flashes produced a discrepancy between the relative retinal locations of the stimuli and their relative physical (i.e., veridical) locations. One version of the cancellation theory would predict that the position of the eye at the time of each flash would be taken into account, thereby producing veridical localization of the stimuli.

It was found, however, that for the time intervals used (up to 300 msec) the judgment of relative location was made predominantly on the basis of the retinal locations of the stimuli. This indicates that the actual position of the eye at the time of each flash was largely ignored for the purposes of this judgment.

Experiment III was identical to experiment II, except that much longer time intervals (up to 1700 msec) were used. A different means of monitoring the eye movement was necessary. As the time interval increased, it was found that judgments of relative location tended to be more veridical.

The results of experiment II are taken as showing that the perceived motion of a pursued object is not due to its being seen in different locations at different instants of time. Rather, as the "motion cancellation" theory assumes, the oculomotor signal is a pure motion signal which "adds" motion to any object which is stationary with respect to the moving retina. However, as shown in experiment I, this motion signal does <u>not</u> cancel the perceived motion caused by image displacement over the retina during pursuit. It is concluded that some other process must be responsible for the perceived stability which occurs during the saccade.

If the eye follows a moving object in an illuminated environment, the background appears to move in the opposite direction ("Filhene's illusion"). This motion is, however, paradoxical; the visual objects appear to move, yet they do not change location. In a sense, then, the background is stable even during pursuit, but it is assumed that this stability is caused by " higher-order factors".

In view of these results it is argued that a "suppression theory" is preferable to the generally accepted "cancellation theory" as an explanation for perceived stability during the saccade. The suppression

theory assumes that ordinarily the retina serves as a frame of reference for motion, but that during the saccade this function of the retina would simply be suppressed. Further support for this theory comes from the recent experimental results of Hans Wallach, who showed that small image displacements of the visual scene go unnoticed during the saccade.

Symbolic Mathematical Laboratory - William A. Martin

A large computer program has been developed to help applied mathematicians solve problems in non-numerical analysis involving tedious manipulations of mathematical expressions. The mathematician uses typed commands and a light pen to direct the computer in the application of mathematical transformations; the intermediate results are displayed in textbook format so that the system user can decide the next step in the problem solution.

This work has extended over several years. During the past year the various pieces of the system were assembled and the whole was tested by solving three problems selected from the literature. A Ph.D. thesis was presented (see Martin, Appendix B) and a movie of the system in operation was completed.

At present, a monograph describing work in this area is being written in collaboration with Joel Moses.

SIN — A Symbolic INtegrator - Joel Moses

Improvements and extensions have been made to the work on symbolic integration reported in Project MAC Progress Report III. (See MAC-M-327, Appendix A.) The program, dubbed SIN, can now integrate all of the problems attempted by its famous predecessor, SAINT (written by Slagle in 1961).

SIN, which is generally more powerful than SAINT, and frequently two orders of magnitude faster, can integrate problems such as

$$\int \frac{\sqrt{A^2 + B^2 \sin^2 x}}{\sin x} dx$$

and

$$\int (1 + 2x^2) e^{x^2} dx$$

A number of methods for solving first-order, first-degree ordinary differential equations have also been programmed. These methods call SIN to solve the integration problems which arise in the solution process. Two examples solved by these methods are

$$xdy - ydx = 2x^2y^2\,dy, \quad \frac{2}{3}y^3 - \frac{y}{x} = C$$

$$x \cos y\, y' + \sin y + \sin x = 0, \qquad x \sin y - \cos x = C$$

The research which is near completion will be presented as a doctoral dissertation, and as part of a book on symbolic mathematics being written in collaboration with William A. Martin.

<u>Manipulations of Algebraic Expressions</u> - Joel Moses

Current methods of factoring a polynomial with integer coefficients can be very time-consuming. In MAC-M-345 (see Appendix A) we present a technique which frequently simplifies the problem of factoring a polynomial by factoring an integer, albeit a large integer.

If P(x) is a polynomial of degree n with Ra bound for the absolute value of its root, then for an integer a, $a > R \geq 1$, $|P(a)| \geq |a_n| (a-R)^n$, where a_n is the leading coefficient of P(x). For P(x) to possess a polynomial factor of degree k, then P(a) must possess an integer factor in the interval $\left[(a-R)^k, |a_n| (a+R)^k\right]$.

Let P(x) have its roots bounded by $R_p \geq 1$, and Q(x) have its roots bounded by $R_q \geq 1$. Let $R = \min(R_p, R_q)$; then, for $a > R$, if the greatest common division of the integers P(a) is less than (a-R), P(x) and Q(x) are relatively prime.

Improvements in the bounds presented in MAC-M-345 were communicated to us by Professor George Collins of the University of Wisconsin.

<u>A Study of the On-Line Computer-Aided Generation of Animated Visual Displays</u> - Ronald M. Baecker

Research has begun on developing an on-line system for computer-aided generation of animated visual displays. Two problems fundamental to this work have been isolated; their investigation was begun during the academic year 1966-67 and will be continued during the academic year 1967-68.

Movement in a picture may be specified by the waveforms of picture parameters expressed as a function of time. A waveform editing capability will be developed for this purpose.

Any interactive computer graphics system must provide a "data structure" with which a user and the system can communicate about pictures. The approach to be investigated is that of the representation and definition of pictures by a set of picture entities with continuous and discrete properties.

The work will culminate in the implementation on the Lincoln Laboratory TX-2 computer of an on-line animation system, and will be described in full in the author's Ph.D. dissertation.

The Problem of Computational Efficiency in Searching Chess-Like Trees - Burton H. Bloom

Research work was performed on problems of computational efficiency associated with heuristic tree-searching problems. The game of chess was selected as a suitable paradigm for the kind of tree-searching problem to be explored. Consequently, as part of the research, a computer program was written to play chess, providing a means for empirically testing the theoretical conclusions about computational efficiency. These conclusions include a description of an optional tree-searching algorithm developed from the abstract analogies of tree-searching processes. A second program for exploring abstract models was written to demonstrate certain properties of the optional algorithm by Monte-Carlo methods.

As of this writing, the experiments with the chess program have been completed. The Monte-Carlo experiments with the second program should be completed by the end of June 1967.

A Heuristic Checker-Playing Program - Arnold K. Griffith

Various aspects of computer checker playing are being investigated in order to produce a complex heuristic program. This program will propose one or a few alternative moves in a given checker situation, without relying on extensive tree search. Such a program, and limited "look ahead" should provide a checker player which is less prone to the type of errors inherent in previous schemes.* A series of investigations into the general nature of the problem have been carried out (under the supervision of Professors Marvin L. Minsky and Seymour A. Papert.)

---

* A. L. Samuel, "Some Studies in Machine Learning Using the Game of Checkers", IBM Journal of Research and Development, July 1959, pp. 211-229

The alpha-beta theorem* was extended to the more general, and more realistic, case of search guided by an imperfect measure of the strengths of alternative situations. Two results emerged. First, a randomly guided alpha-beta search for a minimax variation is considerably more efficient than an exhaustive search. Secondly, a strength measure on the order of accuracy of that employed by A. L. Samuel is not nearly adequate to realize on the immense search time decrease guaranteed by the alpha-beta theorem, which assumes a perfect strength function is used to guide the search. The second result supports the need of a highly selective alternative generator, despite its slowness.

Also investigated was the shape of move trees of depth 5 from plausible checker situations. There was a surprising variability in the number of terminal nodes in such trees. It was further noted that the number of distinct positions to terminal positions of such trees was only half the number of paths, reflecting the high probability that terminal positions may be arrived at by more than one sequence of moves. It is believed that this redundancy increases with greater depth, but further investigation is precluded by computation time limitations.

A rather simple strength evaluation function was developed, based on a statistical analysis of transcribed checker games, and was found to be as good as the best previous efforts, according to findings of A. L. Samuel. This seems to indicate an inherent limitation of schemes of the latter type, rather than speak well of the former type. This result further supports the desirability of a heuristic program over an arithmetical type for alternative evaluation.

Finally, a heuristic program is being developed to investigate possible forcing combinations from a given situation. The results so far have been reasonably successful.

Experiments and Theorem-Proving in Group Theory - Gerald P. Spielman

Two projects have been undertaken. Additional work on the first described project has been postponed for the present, while the second is under current investigation.

---

* Hart, T. P., and Edwards, D. J., The Tree Prune (TP) Algorithm, Artificial Intelligence Project Memo 30, December 1961

W. D. Maurer wrote 84 subroutines in MAD and FAP for the IBM 7094 CTSS, collectively called ALgebra-II. They manipulate structures such as finite semigroups and ideals, groups, subsets, and maps. His work is reported in MAC-M-246 (6/14/65) and MAC-M-282 (12/1/65). I have undertaken to develop a similar system in the LISP language. Experiments employing this system will be made on the PDP-6 with the large memory (256K word core) when appropriate ARRAY structures are available. The programs could also be used as part of a counter-example generator within a theorem-proving system.

At present, working programs include those which can find all subgroups of a group, all ideals (of any specified type) of a semigroup, all subsemigroups of a semigroup, and all subgroups of a semigroup. In addition one can find the smallest subgroup of a given group generated from a given set of elements (the same for semigroups and ideals). One can test for, or discover, certain properties of these structures, e.g., zeroes, identities, inverse elements, associativity, commutativity, and normality of the appropriate algebraic structures.

Routines are anticipated which will handle

1. homomorphic maps between structures,
2. algebraic structure on sets of maps, and
3. generation of specific types of groups, ideals and semigroups.

The second project is the design of a heuristic theorem prover for elementary group theory (See ADEPT: A Heuristic Program for Proving Theorems of Group Theory, by Lewis M. Norton, in Project MAC Progress Report III for an analysis of work on this subject.) The chief aim of the current project is to design a system which understands its subject matter well enough to carry out the non-trivial proof procedures required for interesting theorems in Group Theory. To do this, the range of concepts allowed has been extended to include simple number theoretic statements about orders of sets and elements.

An attempt is being made to facilitate what might be called "global plans" for solving subproblems. The program will decide what hypotheses are relevant to a given conclusion by comparing their implications to the conclusion's. A plan may then be selected which facilitates closing this gap. Subroutines which are "expert" at certain elementary transformations will be available so that useful relationships, which are not explicit from the particular expression of a statement, may be successfully discovered and applied.

Computer Aids to Musical Analysis - Stephen W. Smoliar

There is now available on both the PDP-6 and the IBM 7094 a set of LISP functions for applications in musical analysis. Musical themes are developed by means of various transformations, and we are planning a system which will detect the following transformations of pitch and rhythmic sequences:

I. Fixed Rhythm (pitch sequence altered)

   A. Transposition (both Tonal and Strict)

   B. Inversion (both Tonal and Strict)

   C. Retrogression

   D. Octave Displacement

   E. Triadic Sequence (dependent on harmony)

   F. Scale Patterns

   G. Melody Rotated (about rhythmic pattern)

   H. [illegible]l Expansion and Contraction

   I. [illegible]inations of Techniques

   J. [illegible]rent Techniques (applied to individual measures)

II. Fixed Pitch Sequence (rhythm altered)

   A. Augmentation

   B. Diminution

   C. Duple-triple Alterations

   D. Equal Pitch Duration

   E. Retrograde

   F. Reduced to Fundamental Beats

   G. Ornamentation

   H. Combinations of Techniques

III. Combinations of Both Technique Classes

Currently we have functions capable of detecting the pitch transformations. We also have a function to compute interval vectors for set-complex analyses as proposed by Allen Forte. (See AI Memo 129, EUTERPE: A Computer Language for the Expression of Musical Ideas, April 1967.)

EUTERPE: An On-Line Computer-Music System - Stephen W. Smoliar

EUTERPE is a musician-machine communication system through which a composer may prepare a score of electronic music with as much facility as preparing an orchestral score with the aid of a piano. While the current version of EUTERPE uses a PDP-6 to produce the final audio output, the system may be adapted to any array of electronic music equipment. The major improvement of EUTERPE over previous computer music systems is that it runs in real-time; the user receives his results immediately, and he may correct his errors through an on-line debugging system.

BLANK PAGE

BIOLOGY DEPARTMENT

Molecular Model Building

Computer Display of Protein Electron Density Functions

Converting Atomic Coordinates into a Visual Display

Academic Staff

C. Levinthal

Non-Academic Research Staff

C. D. Barry	E. Meyer	K. Nagano

Research Assistants and other Students

D. Avrin	L. Seldon	M. Zwick

Programming and Filming

L. Feskens	T. Warner	S. Wenz
S. A. Ward		

Molecular Model Building - Cyrus Levinthal

Since the last report several changes have been made to the system used to display both predicted and solved molecular structures. For hardware, a PDP-7 computer has been used as a satellite to the 7094. This has resulted in a system which can call upon either the ESL system or the PDP-7 to produce the visual display. The advantages of the PDP-7 are 1) an increase in the complexity, as measured by the number of vectors composing a picture, of the molecular models that can be displayed; 2) a degree of independence from the 7094 as far as rotation generation of the display is concerned, once the initial 3D-vectors have been generated and transmitted to the PDP-7 to be saved in the core; and 3) the ability to save, on DEC tapes, the set of 3D-vectors required to produce a rotatable picture ready for use with only the PDP-7.

On the software side the opportunity has been taken to make available to the user the possibility of manipulation of known structures such as lysozyme and myoglobin which were previously displayable only in fixed form from coordinate data.

The added space requirements of the system have been met by splitting the previous "package" into specialized segments which are then entered sequentially under control of the user by means of a chaining technique. This removes the necessity of having unused subsections of the program occupying space in the machine core. This structure is ideally suited for incorporation into the overall package of closely related but highly specialized subprograms. The present organization of the programs should make them suitable for simple rewriting when Multics is operating.

Investigations along the lines set out in the previous report have been continued. A detailed investigation of the inter-molecular interactions within the myoglobin crystal has been made using the display capability to show parts of more than one myoglobin molecule as they lie in the crystal. Enumerations of the principle interactions within the lysozyme molecule have been made and a similar investigation has been started for myoglobin. These observations are being correlated with the results from a series of programs designed to try to discover correlations between the amino acid sequence and the structure of proteins.

Work on the predictions of a structure of cytochrome C from knowledge of the chemical sequence and other chemical evidence has reached a point where it has been necessary to consider the question of uniqueness of the proposed structure. To resolve this question it would seem necessary to select from "possible" structures a "best" structure. Two possibilities seem open to us. One involves using partial evidence obtained from x-ray crystallography. The other would use an energy criteria.

In general, before a protein structure is said to be solved it has been necessary to obtain data corresponding to the construction of a 2Å resolution electron density map. This requires accurate observation and detailed analysis of some 10,000 spots on a film representing the scattering of a beam of x-rays by the atoms of the protein arranged in the crystal lattice. For several structures it has been found that while data for a thousand or so of these "spots" is obtainable it has not been possible to collect all the data required for the 2Å resolution map. Rather, the data is sufficient for the construction of a 5Å map. This represents only a "partial" solution of the problem, although the intensities of the spots obtainable may themselves be known to a high degree of accuracy.

The possibility then exists that fusing together the techniques of model building with the information contained in the low-resolution density maps could result in a unique structure consistent both with the known chemical sequence and low-resolution density maps. As a first step in this direction, programs have been written to produce a contouring of a function whose value is known throughout a region of space. Such scalar functions are analogous to clouds of steam or fog. The absence of the opaque surfaces which we rely on when observing the natural world would make visualization of these objects difficult and determination of special relationships almost impossible.

The aim of the display is to provide the viewer with a set of constant density surfaces which allow visualization of the scalar function in terms of the discrete opaque surfaces of the visual world.

To simulate display of these surfaces of constant density, a method has been adopted of contouring the three-dimensional array in two or three orthogonal sets of parallel planes, each set being perpendicular to one of the three sets of axes of the data array. These two or three sets of contours, of a single density value, are displayed simultaneously. The psychological effect of the display is a convincing portrayal of a surface of constant density even if just two sets of orthogonal contours are used. The contouring operation itself is performed upon a 2-D plane of data selected from the 3-D array. These methods have been applied to the display of the electron density function for myoglobin that J. C. Kendrew and H. Watson obtained at 4Å resolution and kindly made available to us.

Future plans are aimed at incorporating these programs into the model building package – via the procedure of chaining – to enable the operator to use the density maps as a guide to the configuration his model should adopt. The possibility of making the process largely automatic, requiring the matching of the model to a set of guide points, is also under investigation. The major step here lies in the prediction of suitable guide points from the density maps, for the refinement of the computer-generated model to minimize the distance between selected points in the

model or between the model and a second structure is already an integral part of the model-building package.

A more direct attempt at obtaining information from the observation of a limited number of spots in an x-ray diffraction experiment by looking for definite substructures, such as alpha-helices in the data, has been tested and initial results with myoglobin seem very promising. In addition, programming has been started to facilitate the conversion of x-ray scattering data, collected on film and then digitized using a flying-spot scanner, to a form directly usable by analyzing programs.

While the main direction of research has been directed toward large protein molecules, the display potential lends itself well to the visualization of small molecules. Programs have been developed to allow these to be displayed from data giving atom coordinates and details of covalent bonds. These programs have also been extended to show the packing of these molecules into the crystal lattice.

It is hoped that this process can be largely automated and will allow for the establishment of a visual library of organic compounds. It is also hoped that it will be possible to display two different molecules simultaneously and be able to change their relative positions. This would enable the interaction between a particular molecule and an associated molecule to be studied.

Throughout, constant reference has been made to the visual link involved in the interaction between the researcher and the computer. Because of this, some attention has been paid to considerations of the presentation of the visual results of the work. The essential need for a 3-D representation of the objects studied has led to a study of the use of cine films, vectographs (3-D prints produced by a Polaroid technique on a single piece of film, and rendered three-dimensional by using polarizing glasses for viewing), and the production of a rotatable display in conjunction with the PDP-7.

Computer Display of Protein Electron Density Functions - David Avrin

Under investigation have been the possibilities of using a three-dimensional display system, such as the KLUDGE, or the PDP-7, at the Science Teaching Center, for aiding in the determination of protein structure from the electron density distribution of the protein crystal. The electron density function, which we are attempting to visualize, is a three-dimensional scalar function obtained from x-ray crystallography studies. By extension from the case of two-dimensional scalar functions, where one draws lines of constant density, it should be possible to visualize the three-dimensional case by displaying contour surfaces of constant density. However, it is difficult to model and display such a surface.

I believe that we have found a promising solution to this difficulty. The psychological effect of our displays is a convincing portrayal of a surface of constant density. In particular, with regard to protein displays, it appears that we also have a helpful tool for simplifying the transition from the x-ray data to the molecular structure.

Our method found its origin in the traditional means used in x-ray crystallography for visualizing data. However, we have added a few additional features. These features are only possible when using an on-line, computer-driven display system.

We have found that if any two of the three orthogonal sets of parallel planes defined by the array are line contoured, and these two sets of contours displayed simultaneously upon a device such as the KLUDGE or the PDP-7 at the Science Teaching Center – using rotation to simulate a three-dimensional display – a remarkable perception of a surface of constant density can be achieved. This display is almost free of any directional constraints or favoritisms. Adding the third set of contours to the display removes these small remaining effects. However, these marginal improvements over using just two sets of contours are not worth the sacrifice in display space.

Even when creating such a display, the problem of following the peptide chain in a unit cell of protein crystal remains an insoluble problem. However, by using the light pen we have created two additional aids which hold considerable promise for both simplifying this particular aspect of protein structure investigation, as well as for displaying protein structure for teaching purposes.

First, the light pen is used to identify the location of a particular contour or group of contours in a subsection of the unit cell displayed upon the KLUDGE. This identification initiates a function which examines the compiled orthogonal contour list of the entire unit cell. This function performs a search and assembles a contiguous surface of constant density attached to the contour identified with the light pen. By judiciously choosing the density level to be contoured, this function appears to have the capability of following and displaying reasonably long sections of the protein polymer.

Second, the light pen is used as a drawing tool for threading a string through a particular region of the density function which is believed to contain the peptide chain. The interesting feature of this use of the light pen is its ability to sketch in three dimensions, rather than in two. Hopefully, this manually drawn pathway through a prepared display of a section of the polypeptide will provide the three-dimensional co-ordinates for the approximate pathway of the peptide chain.

In summary, it should be added that many features of these programs have application to any investigation of a three-dimensional scalar function.

We are continuing to investigate and expand the power of these methods. Most of our programming is in MAD, using library routines developed by and for the Biology Group for various purposes. This summer we hope to create a rough model of the protein myoglobin by using the drawing techniques and path-following algorithm mentioned above. Also, we hope to integrate these procedures with the protein package being developed by the Biology Group. Figure 4 shows two views of approximately a 15 Angstrom piece of peptide chain in an unidentified section of myoglobin (data supplied by J. C. Kendrew and H. Watson, Cambridge, England). The display is much more convincing when rotated in real-time on the KLUDGE or PDP-7.

Converting Atomic Coordinates into a Visual Display - Edgar Meyer

A large number of chemical compounds have been investigated crystallographically and their structures are reported in the literature. A MAD program has been written to use either KLUDGE or PDP-7 display routines to draw the desired molecule. A chemical bond is represented by a line connecting two atoms. Chemically non-bonded atoms in the sequence are connected by invisible lines. A three-character name can be given to each atom. The program is made to generate the requisite crystallographic symmetry operators and display either the molecule by itself or the contents of one unit cell. The required input includes the cell constants, atomic names and coordinates, a number specifying the space group, and, at present, a hand-made connectivity table.

Rotating the display about three orthogonal axes gives the viewer a perception of depth so that the model appears to be three-dimensional. This effect is enhanced when the contents of a unit cell are drawn by displaying the outline of the crystallographic unit cell. By manipulating the display, the viewer may gain perceptual insight into fairly complicated molecular structures.

The symmetry information is contained in a subfunction and additional space groups may be included as required. Specific rotational information may be input to generate stereo pairs.

At present the major limitation to the general application of the program is the necessity of reordering the sets of atoms to fit a connectivity table made with the object of minimizing the number of invisible lines that must be drawn. Thus a major extension of the program would be to devise an efficient link to an existing library of chemical and connective information.

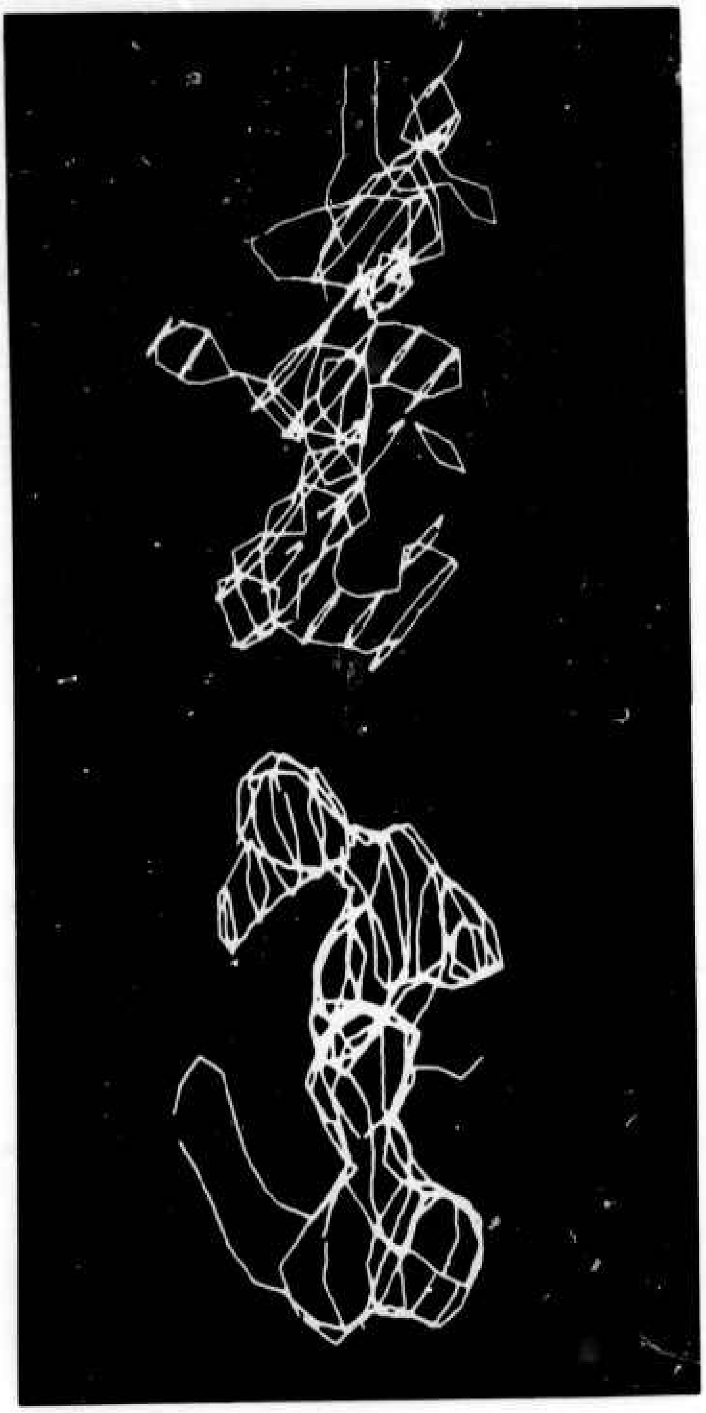


Figure 4. A 15Å Piece of Myoglobin Peptide Chain

In addition, the program could be extended to draw spheres of specific radii around the various atoms, giving an idea of atomic and molecular packing. Alternatively, when the information is available, ellipsoids of vibration could be drawn for the individual atoms. Likewise, the light pen could be used to activate certain subfunctions that would produce numerical information about the geometry of the molecule: bond lengths, bond angles, torsion angles, and equations of the best-fit planes defined by specific atoms, together with deviations from these planes.

With the advent of greater display power, a useful addition would be the possibility of displaying the contents of any of the twenty-six neighboring unit cells, so that atomic and molecular interactions could be investigated under operator control.

This program has been successfully displayed on a PDP-7 that is linked by a telephone line to Project MAC. It is hoped that, with increased computer availability, such a graphical display would have more general use as a quick and powerful tool for the study of structural information.

BLANK PAGE

CIVIL ENGINEERING DEPARTMENT

Computer Systems Conference

Time-Sharing Research in Soil Mechanics

Academic Staff

L. N. Beckreck	A. M. Hershdorfer	G. M. Sturman
M. J. Biggs	W. H. Linder	J. M. Sussman
J. T. Christian	R. D. Logcher	R. A. Walter
A. G. H. Dietz	W. M. Pecknold, III	R. V. Whitman
A. J. Ferrante	F. E. Perkins	
A. T. Foster	D. Roos	

Non-Academic Research Staff

R. L. Daniels	R. V. Goodman	J. C. Jordan

Research Assistants and other Students

A. J. Bodhe	R. I. Karash	L. C. Teague, Jr.
S. D. Chan	R. E. McPhail	
E. J. Hall	B. Schumacker	

Computer Systems Conference - Frank E. Perkins, Thomas A. Asselin

A workshop conference on Civil Engineering Computer Systems was held at M.I.T. during the last week of July 1966 under the auspices of the Department of Civil Engineering and the Center for Advanced Engineering Study. The purpose of the conference was to demonstrate and evaluate, by the participation of those attending, recent advancements in hardware and programming technique which have made possible more complete integration of the computer into the engineering design process. The conference participants included 35 academic people and 77 participating engineers from private consultants and government agencies.

The primary emphasis of the conference was directed toward the Integrated Civil Engineering System (ICES) which is currently under development in the Department of Civil Engineering. Both the systems programming and engineering application aspects of ICES were presented by means of general lectures and workshop sessions. In the latter, the participants, working in small groups, developed their own application subsystems for ICES and solved engineering problems via the existing subsystems.

In addition to ICES, other topics of potential value to the design engineer were considered; these included graphical input/output and time-sharing. The following is a brief report on how the subject of time-sharing was presented and the results which were achieved, using Project MAC facilities.

The staff of the Civil Engineering Systems Laboratory is in complete agreement with the idea that time-sharing offers an extremely powerful mechanism for integrating the computer into the design process. The assumptions that time-sharing will continue to develop rapidly and that this mode of operation will become more commercially attractive to engineering users have been made throughout the ICES development. Hence, we felt a definite obligation to ensure that the conference participants understood the general concepts of time-sharing and its implications to the engineer.

Although many of the participants had some established ideas about time-sharing, their information was generally minimal and sometimes erroneous (e.g., several persisted in confusing remote computing with time-sharing.) A few of the academic people had made use of time-sharing systems at their own schools or had closely followed the related professional literature. At least two of the consulting firms which were represented had made some use of the limited commercial services which were available to them. However, these were the exceptions; taken as a whole, the participants had a very limited understanding of time-sharing.

Considering the importance of time-sharing to ICES, and the participants' lack of experience with such a system, we decided that a simple demonstration of the system would be inadequate for our purposes: instead, "hands-on" experience by each participant would be essential if a feeling for this mode of operation were to be achieved. Furthermore, it was felt that this experience must consist of something more substantial than a brief, "canned" session at the console. To be successful, the experience should include a session of real problem-solving with significant amounts of man-machine interaction. In conflict with this desired approach were the limitations of time and facilities. With four consoles each available for a total of 18 hours (3 days at 6 hours/day) there was less than 40 minutes of console time available to each of the 112 participants. This was hardly enough time for a meaningful exercise. A compromise solution was reached by organizing the participants into 24 workshop groups each consisting of 4 or 5 individuals. Each group was then assigned to a console for a full 3-hour work session. The actual typing at the console was rotated among the group members but all of the members were able to observe the computer responses and thus participate continuously in the activity.

In order that the workshop sessions might be devoted entirely to actual use of the system with a minimum of lecture time, an introductory lecture and demonstration was presented to the entire group on the first day of the conference. This presentation was made by Professor J. M. Biggs, Director of the Civil Engineering Systems Laboratory, at the Kresge Little Theatre and lasted for about one hour. Professor Biggs discussed the philosophy of time-sharing, emphasizing its potential for on-line design, and described some of the important hardware and software aspects of the M.I.T. Compatible Time-Sharing System (CTSS). In his demonstration, which was presented as an integral part of the lecture, he described some of the system's editing and information-retrieval capabilities while carrying out a simple structural design using the STRUDL subsystem. Response from the MAC system was quite good throughout the demonstration, thus contributing greatly to the interest shown later at the workshop sessions. However, a point worth noting for future consideration is that 23-inch closed-circuit television screens are not completely adequate for displaying the text of console statements.

For the workshop sessions it was possible to provide adequate working space and seating room for all the group members at each console. This was quite important; without a close but comfortable viewing point the non-typing members of the groups could easily have lost the feeling of interaction which time-sharing provides.

The workshop sessions were conducted by Professor F. E. Perkins and Mr. T. A. Asselin of the Civil Engineering staff, each man providing instruction simultaneously for the two workshop groups in his classroom.

Each session participant was provided with a time-sharing guide which provided a minimal amount of reference information. Initially, the instructors led the groups rather pedantically through the procedures of logging in, listing and printing files, and then program generation, editing, and execution. The amount of guidance offered by each instructor decreased rapidly, until at the end of about one hour the groups were quite effectively operating on their files and were freely experimenting with other administrative commands.

The two remaining hours of each workshop were devoted to engineering problem solving using the time-sharing versions of COGO and STRUDL. Two sample problems are shown in Figures 5 and 6. To facilitate solution of the structural problem, the necessary descriptor data file had been established by the instructor prior to each session. Since no lengthy data file was required for the geometry problem, it was possible in both problems for the groups to launch almost immediately into the solution process without extraneous preliminaries. Again, the instructors actively led the groups for a short period of time, but quickly assumed the role of advisors and provided assistance only as needed.

Although most of the participants had only a limited knowledge of COGO and STRUDL, they were usually able to carry the solution of both sample problems to a reasonably satisfactory point before the end of the three-hour session. Therefore, it was possible to introduce other commands and programs which were of interest to the groups. These included linking to other files to demonstrate communication between users, using RUNOFF to print parts of the time-sharing manual as an example of information retrieval, and operation of the DOCTOR program both to illustrate a sophisticated symbol manipulator and to demonstrate the high degree of man-machine interaction which time-sharing makes possible. (See Stone, this volume.)

Response of the participants to the time-sharing workshops was generally very satisfactory, with many coming away quite excited about the potential which this mode of operation would provide in their own activities. The participants were very much interested in and concerned about the problems of file security, overhead time, degree of availability, and response time in a commercially available service. They seemed to be in general and decisive agreement that they would not make extensive use of a commercial service unless it were developed to the point where accessibility and reasonable response were essentially guaranteed.

The success of the workshops was due in large measure to the fact that four party lines to the MAC system had been assigned for the duration of the conference. The system appeared to be in great demand with a full complement of users found to be on the system at almost every inquiry. It is certain that without four party lines we could not have

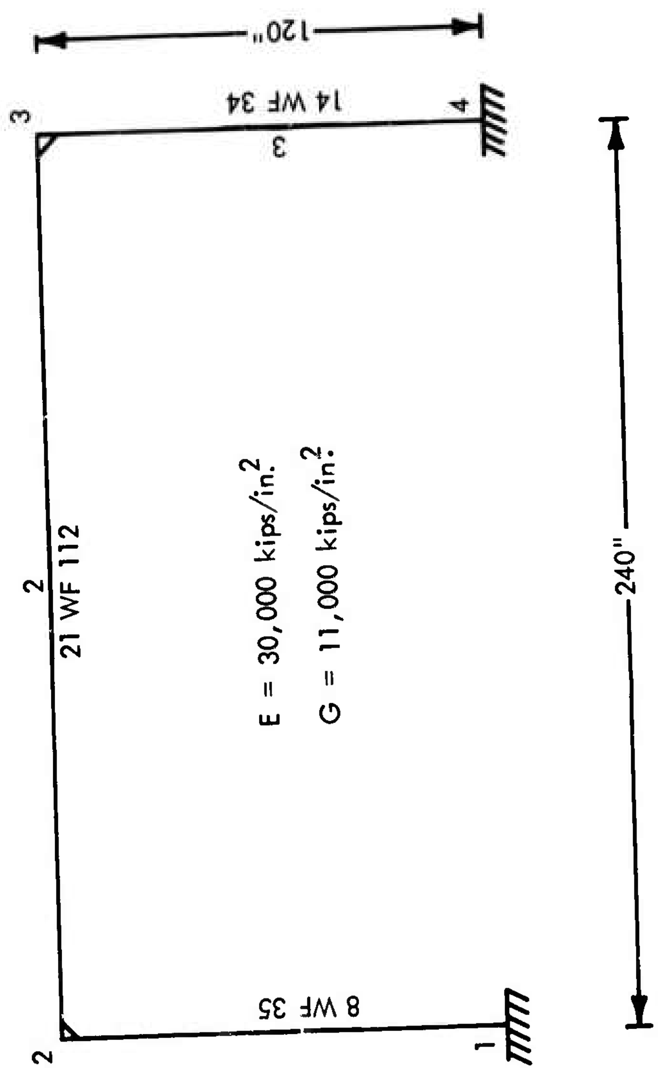


Figure 5. A STRUDL Problem

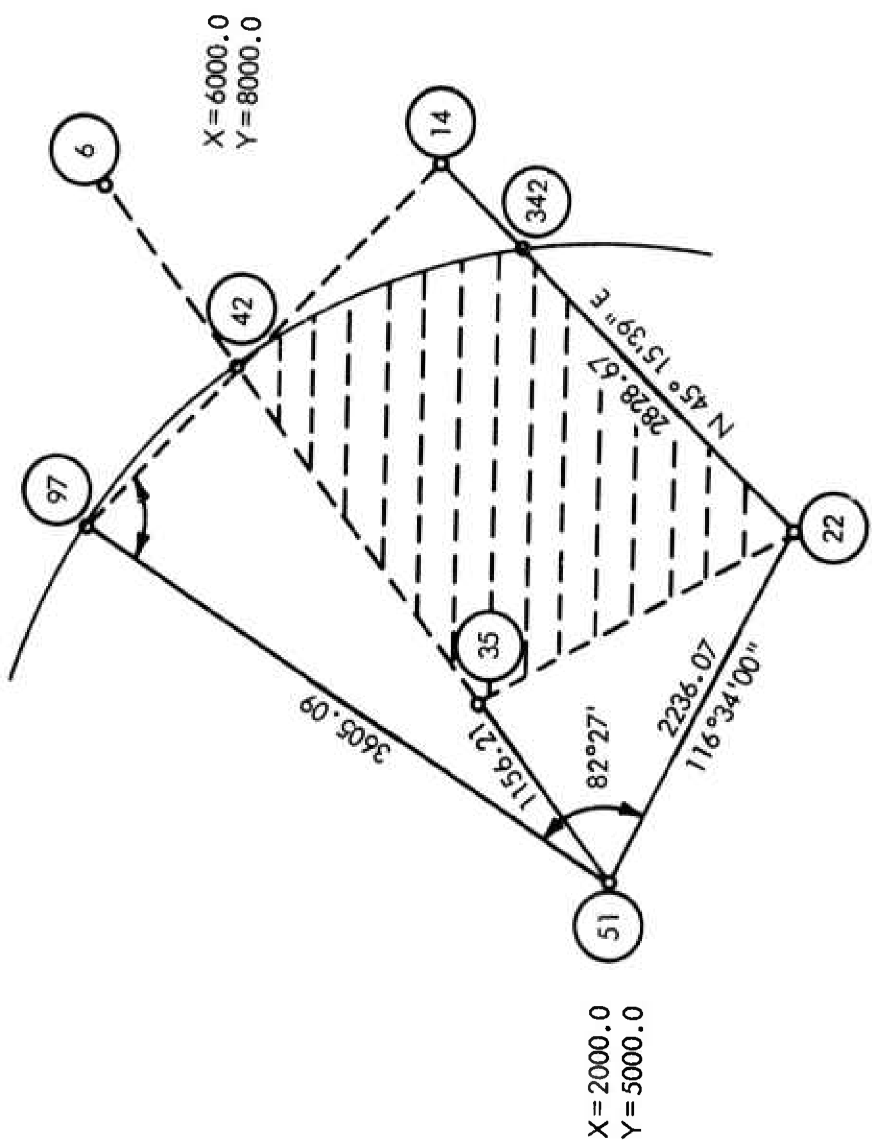


Figure 6. A COGO Problem

maintained four operating groups for any significant period of time. The only significant problems arose on the third day of the workshops. The MAC system was down throughout the entire morning session. A voluntary evening workshop was substituted for the morning session and was attended by several people. During the afternoon session on the third day the MAC system was again operational, but response was exceedingly poor. In fact, the response to some of the STRUDL commands was so slow that the structural aspects of the session were soon abandoned. On the other two days the response was satisfactory.

It is interesting to note that the total amount of computer time used by a group during its three-hour session varied from a low of 0.8 minutes to a high of almost 5.0 minutes, with an average use estimated at 2.5 to 3.0 minutes.

Perhaps the most significant thing gained by this experience was confirmation of the belief that time-sharing has tremendous potential for engineering design. Also significant is the fact that effective use of the system could be made by engineers after only a very brief period of indoctrination. However, it was apparent that effective use depends entirely on the availability of adequate software. System capabilities such as those provided by CTSS, coupled with design-oriented, interactive systems like COGO and STRUDL, are necessary for successful engineering use.

<u>Time-Sharing Research in Soil Mechanics</u> - Robert V. Whitman, John T. Christian, Kaare Hoeg, William A. Bailey, and Robert E. McPhail

Research during the reporting period was concentrated in two areas of engineering significance: slope stability and seepage. The slope stability programs, BISHOP and MGSTRN, were written during the academic year 1965-66; major efforts in this area included testing the programs and using them in engineering situations, as well as analyzing several real problems. These analyses revealed necessary modifications of the programs, and these changes are still being made as the need becomes apparent. Use of the programs has permitted a significant increase in understanding the mechanism of slope failure. Some conclusions and new questions are presented in a paper by Robert V. Whitman and William A. Bailey, "Use of Computers for Slope Stability Analysis," presented at the A.S.C.E. Specialty Conference on Stability and Performance of Slopes and Embankments, Berkeley, California, August 1966, and soon to be published by A.S.C.E.

The seepage research continues from academic year 1965-66 and is jointly sponsored by the Massachusetts Department of Public Works (Contract Number 1782). The work has involved extending the present program for confined flow to handle seepage with a phreatic surface,

which must be found by iteration. At the same time, anisotropic permeability has been introduced. A further change has been to make the input as compatible as possible with the slope stability programs so that these can be united eventually. Work has progressed well and should continue through the summer of 1967.

Both programs have been used in teaching. In particular, the stability programs have been valuable educational tools in a special graduate course on slope stability analysis by computers (Course 1.39, Fall 1966) and in undergraduate soil mechanics (Course 1.06, Spring 1967).

BLANK PAGE

COMPUTATION STRUCTURES

Table-Driven Compiler Systems

Design of Sequential Machines with Fault-Detection Capabilities

Effects of Scheduling on File Memory Operations

Resource Allocation

Asynchronous Computational Structures

An Abstract Parallel Processing System

Checking Sequences

Academic Staff

J. B. Dennis	F. C. Hennie, III	C. L. Liu
J. J. Donovan	Z. Kohavi	D. F. Martin
E. L. Glaser	H. B. Lee, Jr.	C. E. Shannon

Research Assistants and other Students

A. A. Bushkin	P. G. Hebalkar	A. Scholl
J. R. Carbonell	H. F. Ledgard	D. R. Slutz
P. J. Denning	F. L. Luconi	P. Vale
G. Falk	R. L. Martin	
J. A. Hamilton	S. S. Patil	

Table-Driven Compiler Systems - Chung L. Liu and James A. Hamilton

The results of a Table-Driven Compiler System, designed and implemented on the CTSS system, were presented at the 1967 Spring Joint Computer Conference* and will be published as a Project MAC technical report. Further work in our study of table-driven compiler systems includes:

1. Design and implementation of a general-purpose table generator (a general-purpose bootstrap compiler). Such a generator will produce driving tables for the Table-Driven Compiler System.
2. Implementation of a subset of ALGOL 60. Our goals are to evaluate the advantages and disadvantages of the compiler system and to compare it with both table-driven and other systems.
3. Preliminary study of the design of a similar system for Multics.

It is hoped that the experience gained in our work in the current system will lead to improvements in the design of the next system.

Design of Sequential Machines with Fault-Detection Capabilities - Zvi Kohavi

Designing fault-detection experiments for sequential machines is a problem strongly related to that of machine identification. A sequential machine M, whose structure and behavior is to be determined, is given to the experimenter as a "black box". An input sequence is applied to M and the corresponding output is recorded. The experimenter's objective is to find a state table which uniquely defines the given machine. A fault-detection experiment is required to demonstrate whether the given unknown machine is indistinguishable from a known one. The design of such experiments is the first objective of current research.

Designing fault-detection experiments for arbitrary machines is extremely difficult, since the resulting experiments are very long and inefficient. Consequently, it becomes necessary to design machines so that maintenance and testing is manageable. As a result of this research,

---

* "The Design and Implementation of a Table-Driven Compiler System", C. L. Liu, G. D. Chang, and R. E. Marks, Proceedings of the Spring Joint Computer Conference, 1967.

a method has been developed for designing sequential machines in such a way that they possess special distinguishing sequences identifiable by very short fault-detection experiments.

A sequential machine for which any input sequence of a specified length is a distinguishing sequence is said to be definitely diagnosable. A method has been developed to obtain for any arbitrary sequential machine a corresponding machine which contains the original one and is definitely diagnosable. Similarly, these techniques are applied to embed machines which are not information lossless of finite order, or which do not have the finite memory property, into machines which contain either of these properties.

Simple and systematic techniques have been presented for the construction and determination of the length of the distinguishing sequences of these machines. Efficient fault-detection experiments have been developed for machines possessing certain specified distinguishing sequences. A procedure has been proposed for the design of sequential machines such that they possess these special sequences, and for which short fault-detection experiments can be constructed.

The next phase of this research will be the generalization of these methods to a larger class of machines, and the design of fault location experiments; i.e., experiments which not only detect faults in the machine, but which indicate the location and nature of such faults.

Effects of Scheduling on File Memory Operations - Peter J. Denning

Secondary memory activity is a prime factor affecting the performance of any computing system, for secondary storage is, in a very real sense, the heart of the system. In a multiprogrammed computing system there is a considerable burden on secondary storage, resulting both from constant traffic in and out of main memory (core memory), and from the large number of permanent files in residence. It is clear that the demand on the file memory system is extraordinarily heavy, and it is essential that every part of the computing system interacting with file memory do so smoothly and efficiently.

Two types of file memory are commonly used: fixed-head types (drums), and moving-head types (disks). For each type of device, the proposition that utilization of secondary storage systems can be significantly increased, by scheduling requests so as to minimize mechanical access time, has been investigated mathematically. In the case of fixed-head drums the proposition is spectacularly true, with utilization increases by factors of 20 easily attainable over conventional first-come, first-served scheduling. For moving-head devices such as disks the proposition is still true, but not dramatically so; optimistic predictions

look for at best a 40 percent improvement, an improvement only attained with serious risk that some requests receive no service at all. There is a workable solution that circumvents this problem — a method of "scanning" the arm back and forth across the disk: under this regime the improvement is but 12 percent. (See Denning, Appendix C.)

On the basis of this investigation and recent work by Nielsen* we feel fully justified in making this statement: moving-head storage devices are entirely inadequate for systems with paged memories. Systems using such devices will almost certainly encounter serious performance degradation. The problem with moving-head devices is that a certain irremovable, significant fraction of time is spent moving the heads, and congestion develops even under light loads. Nielsen, in his simulation of the 360/67 system, has found that the paging disk was a major bottleneck, this despite the existence of eight concurrently operating modules. For fixed-head devices, the story is entirely different, and the mathematics reveal two very interesting facts:

1. As the demand on the device increases, utilization approaches 100 percent.
2. The waiting time of a request depends primarily on the device revolution time and only secondarily on the load.

Resource Allocation - Peter J. Denning

Resource allocation in multiprogrammed computers is deceptive. Process scheduling and core memory management have been discussed extensively. Development of techniques has progressed independently along both these lines. No one will deny that a unified approach is needed. Probably the most basic reason behind the absence of a general treatment is the lack of an adequate model for program behavior. In Project MAC Progress Report III, we reported on preliminary work into the "working set model" for program behavior. This work has been continued, concepts sharpened, ideas refined, so that now we can report on work into the much more difficult area of resource allocation.

We suppose that the computer memory has two levels: main memory and secondary memory. Only data occupying main memory can be processed. We suppose that both levels of memory are paged, the page being the basic unit of information storage and transmission; it

* Nielson, N. R. The Analysis of General-Purpose Computer Time-Sharing Systems, Ph. D. Thesis, Stanford University, School of Business Administration, 1967

requires an average of T seconds to move a page (in either direction) between the memories. T will be called the access time for secondary memory.

Roughly speaking, a working set of information ("working set" for short) is the minimum collection of pages a given process requires present in main memory at any given time to operate "efficiently". Thus there is a one-to-one correspondence between a process and its working set; this leads quite naturally into the allocation problem, for the process and its working set present jointly a demand for processor time and for memory space-time. Various considerations have led to this precise definition of the working set W of a given process:

$$W = \left\{\begin{array}{l}\text{all the pages referenced by the} \\ \text{process during the last T seconds}\end{array}\right\}$$

Here T is the access time for secondary memory. From this definition arises a simple allocation mechanism, one that takes into consideration the presence or absence of working sets in main memory when deciding which processes to run. Nor is the mechanism hasty in removing pages in a working set whose process blocks, be it for console interaction or for acquisition of a new page. A working set must remain in main memory long enough for a typical console interaction to be completed, say 5 or 10 seconds. Obviously this leads to increased demand on main memory; but this is the price that must be paid when consistently good service to the customer is a prime objective.

A demand is a pair (p,w), p being the processor time demand of a process, and w its current working set size. Both p and w are easy to measure:

1. w is increased or decreased by one each time a page enters or leaves the working set, respectively.

2. Each time a process is given a processor, it is allocated a quantum Q (all processes receive the same quantum Q), where Q is chosen large enough so that it is more likely that the process will give up the processor by blocking rather than by quantum runout. In a highly interactive system, Q should not have to be more than 1 or 2 seconds. Suppose the process has the processor for t of its Q seconds, ($t \leq Q$); then demand is

$$p = \min(1, \frac{t}{Q})$$

The demand is a random function of time, but we assume that it is well-behaved in the sense that large changes in demand are much less likely than small changes. Under this assumption, we will base allocation decisions on the prediction that the demand during the next Q seconds of

operation is most likely to be that measured at the end of the last Q seconds.

The concept "demand" leads to that of balance; roughly speaking, the system is balanced when the sum of demands presented by all the running processes just consumes the available resource. All other processes demanding service, but currently not receiving it since their demands would upset balance, are said to be ready. More precisely, balance exists when

$$\sum_{\substack{\text{all running}\\\text{processes}}} (p, w) = (N, M)$$

where N is the number of processors, and M the number of pages of main memory. The object of the allocator is to maintain the system in balance; that is, whenever a process has been blocked long enough for its working set to disappear from main memory, the total demand diminishes, so the allocator must choose one of the next processes ready to run whose demand most nearly restores the system to balance. This can be done iteratively so that scheduling overhead depends not on the total demand, but only on the degree of imbalance.

The balance criterion for allocation represents a unified approach to the problem. These ideas are the topic of my Ph. D. research. The next objective is a mathematical treatment of their behavior, which will yield such information as expected waiting time and bounds on the tails of waiting-time distributions, thereby yielding a mathematical characterization of the system's behavior from the viewpoint of a process.

Asynchronous Computational Structures - Fred Luconi

The recent trend in the design of digital equipment toward increased use of integrated circuitry implies that certain design techniques ought to be re-evaluated. Most design languages reflect the current practice of organizing computer systems design around registers and register transfer gating. Such descriptions are used because they provide a means of expression easily understood both by designers and other parties who must become deeply involved with the system; e.g., maintenance technicians. Since these structures are most often organized according to function, last-minute changes usually can be made on only the functions involved, with little reorganization of the remaining structure required. The Huffman state-table model for sequential machines has found relatively little use in practical design, because structures of reasonable size generally lack this property of understandability when derived from the use of such techniques. Therefore, one criterion for a design representation scheme appears to have been ease of interpretation.

For conceptual purposes it appears necessary to base a design language (digital system representation scheme) on specifying in detail the operation and interconnection of a small set of easily understood design blocks or subsystems. If, in addition, we want functional dependencies, timing, and sequencing constraints between subsystems to be stated explicitly in the network representation, and not by implicit rules of interpretation, then we must allow the specification of such constraints to be made only by declaring communication paths between related subsystems. For this reason it appears advisable to use descriptions of the system building blocks (the subsystems) in which sequences of output values can be determined from sequences of input values without regard for time dependencies between these sequences; unless, of course, timing constraints are explicitly expressed as part of the functional relationship. Since synchronization of separate subsystems can be achieved only through the sharing of common information, e.g., a clock signal, the relative timing of subsystems which are logically disconnected must be regarded as arbitrary. In other words, we should assume the relative timing of elements within a network to be arbitrary and allow the network to operate asynchronously and questions of synchronism to be answered only in terms of interconnection constraints.

Still another important feature missing in existing digital representation schemes is an effective means for dealing with structures in which several different computations may be proceeding concurrently. Machine designers have already dealt with concurrency problems in control and I/O, where hardware elements are multiplexed among many activities. Multi-processor computing systems represent present-generation examples of large structures in which several computations share hardware, and many programming system designers currently are in the process of defining methods for specifying algorithms allowing exploitation of the parallel-processing capabilities of this hardware. If we define a process to be the locus of information exchange and transformation, then it is desirable to be able to describe multi-computation structures which allow several processes to proceed with arbitrary relative timing and with permitted inter-communication.

Questions of determinism are always associated with asynchronous structures. If we define output functionality such that a computational system has its output-value sequence determined uniquely by its initial "state" and the sequence of input values, then we require that, in our representation scheme, procedures exist to guarantee output functionality of a network despite its asynchronous operation. For comparing different realizations of the same function, or for use as a criterion for establishing rules of transformation from one realization of a function to another, it would be advantageous to have a test of equivalence between output functional structures.

Preliminary investigations of asynchronous representation schemes are reported in Project MAC Computation Structures Group Memos No. 25 and No. 27, and the Ph. D. research described here will be available as a Project MAC Technical Report. Initial results of this research will also be found in the Proceedings of the Eighth Annual Symposium on Switching and Automata Theory and in forthcoming Computation Structures Group memos.

An Abstract Parallel-Processing System - Suhas S. Patil

An abstract parallel-processing system which is a further development of the "program graphs" of Jorge Rodriguez* has been studied. (See Patil, Appendix B.) In the new program graph model, directed graphs not only represent a computation to be performed but also define, through stated rules of successive alternation of the graph, the semantics of its execution. Particular emphasis is placed on separating data and procedure information so that programs are represented in "pure procedure" form and can be shared among several concurrent computations. The instructions of the procedures are represented by graphs called instruction-graphs. The initial data of the procedures are kept in graphs called data-structures. The data-structures of a procedure may contain many structured data; in particular it may contain other procedures.

Programs are constructed by nesting program-graph procedures. Execution of a program is started by applying the procedure representing the program to the input data. The computations resulting from different applications of a procedure have their own private data-structures, but all of the computations use the same instruction-graph. The computation takes place through asynchronous and concurrent transformations on the data. The data used by computation can be structured and the unwanted data can be discarded systematically.

The transformation of applicative expressions into program graphs has also been studied. The applicative expressions considered are the λ-expressions.† The program-graphs representing λ-expressions can be used in the parallel-evaluation of the λ-expressions.

---

*J. Rodriguez, Analysis and Transformation of Computational Processes, Project MAC Memorandum MAC-M-301, Computation Structures Group Memorandum No. 22, March 1966

† P. Landin, "A Correspondence Between ALGOL 60 and Church's Lambda-Notation", Comm. of the ACM, Vol. 8, No. 2, February 1965, pp 89-102; and also Comm. of the ACM, Vol. 8, No. 3, March 1965, pp 158-169

Checking Sequences - Donald L. Slutz

Under certain conditions, a single input/output sequence called a checking sequence uniquely characterizes all the input/output transformations of a finite-state machine and thus may serve to check for the correct machine behavior. Since Hennie's synthesis procedure often produces unnecessarily long checking sequences, an attempt was made to determine lower bounds on the lengths of checking sequences and to develop synthesis procedures to achieve these bounds.

A program was written that would take an input/output sequence and generate all distinct N (or less) state machines that would produce the given output sequence from the given input sequence. The program then would not only indicate whether a given sequence was a checking sequence, by generating just the correct machine, but would also indicate "how close" if it was not. A large number of relatively short checking sequences were determined using the program and intuition. They were found to differ greatly from Hennie's sequences in that it was not possible to identify easily what state the machine was in at any particular place in the sequence, and any prefix of the sequence gave virtually no information about the structure of the machine. Interestingly, it also seemed that checking sequences for machines that possessed no distinguishing sequences were not significantly longer than for machines that did.

A number of techniques were attempted to characterize these short checking sequences, but they met with little success. There do seem to exist certain types of machine structures for which short checking sequences can easily be found, even though much of the design is of an intuitive nature.

COMPUTER SYSTEM RESEARCH

CTSS and Multics System Development

A. CTSS Maintenance

B. Use of CTSS for Multics Development

C. Multics Implementation - General

D. Multics Implementation - By Area

E. Specification of Benchmarks

F. Hardware

A Data-Storage Structure for Multiplexed Information Systems

A Communications Bus Machine

Academic Staff

A. A. Bushkin
F. J. Corbató
J. H. Donovan
A. Evans, Jr.
R. R. Fenichel
E. L. Glaser
J. H. Saltzer

Non-Academic Research Staff

G. F. Clancy
M. E. Child
D. Clark
C. A. Cushing
R. C. Daley
S. D. Dunten
C. Garman
R. M. Graham
B. Kernighan
S. Kidd
C. Marceau
K. J. Martin
N. I. Morris
C. D. Olmsted
M. A. Padlipsky
R. Rappaport
D. M. Ritchie
S. L. Rosenbaum
G. G. Schroder
T. P. Skinner
P. Smith
J. W. Spall
M. S. Spier
M. R. Thompson
T. H. Van Vleck
D. B. Wagner
S. Webber
D. R. Widrig

Research Assistants and Other Students

D. J. Edwards
J. M. Grochow
J. D. Mills
M. D. Schroeder
L. L. Selwyn
R. H. Thomas
B. D. Wessler
J. C. Yochelson

Guests

N. Adleman - System Development Corporation
H. J. Herbert - Shell Oil
D. H. Johnson - Department of Defense
P. Schicker - Swiss Federal Institute of Technology
H. L. Stambler - U.S. Weather Bureau
W. R. Strickler - Shell Oil

CTSS and Multics System Development - Fernando J. Corbató

The attention of the system-programming staff has been focused on two major systems, the Compatible Time-Sharing System (CTSS) on the IBM 7094, and the Multiplexed Information and Computing Service (Multics) system for the GE 645. Multics is a joint effort of M.I.T. (Project MAC), Bell Telephone Laboratories, and the General Electric Company. The three organizations are participating in system implementation, and are cooperating in software development. The design philosophy and general structure of the system were discussed in last year's progress report.

A. CTSS MAINTENANCE

During the reporting period, the principal effort of the system-programming staff was directed toward continued development of the Multics system, as is the major portion of this report. As a result, the effort expended on CTSS was confined primarily to routine maintenance activities. Although several new commands were added to the system's repertoire, these were primarily contributions of individual users who had created the programs originally for their own purposes. A small change in the LOGIN procedure was introduced, whereby a user is informed, prior to logging in, of the relative load on the system. When the system is fully loaded, a user does not attempt to LOGIN, and the machine time formerly required to process such abortive LOGIN attempts is saved.

B. USE OF CTSS FOR MULTICS DEVELOPMENT

Besides providing time-shared computing services to the entire MAC community, the CTSS operation has greatly facilitated the creation and debugging of Multics system programs. By using the testing and debugging system (described in last year's report), a Multics system programmer was able to create, edit, and compile his programs on the time-shared IBM 7094, execute them on the GE 635 (subsequently replaced by the GE 645), and then again utilize CTSS facilities to examine the results and, if necessary, repeat the same process again. The importance of this method cannot be overemphasized, for during the reporting period the CTSS EPL (Early PL/I) compiler was the only working compiler available to the Multics development effort. Further, the value of CTSS was evident in those areas which have come to be taken almost for granted by users of time-sharing systems: ease of on-line program creation and source-code editing and immediate feedback from the compiler in the area of syntactical errors. A final, though less obvious, point is that CTSS allowed the three organizations working on Multics (with Bell Laboratories a considerable distance from Cambridge) to work in parallel,

communicate, and share both ideas and results. In short, we could not have gotten as far as we have in Multics development had CTSS not been available.

C. MULTICS IMPLEMENTATION — GENERAL

Since most of the Multics software basic structure design was completed by the end of the previous reporting period, this past year has seen the programming effort increasingly directed toward detailed design, coding, and check-out of system modules. A rough guide of accomplishment may be found in the bar charts of Figure 7, which represent the output of source code pages (and their development status) throughout the reporting period; the phases referred to are defined in Section E. In Figure 8, the state of the Multics System-Programmers' Manual (MSPM) is depicted at the beginning and end of the reporting period. The remainder of this report will attempt to highlight the nature of what was accomplished.

The EPL language, a subset of PL/I, was used as the primary medium for program-writing. The use of this higher-level language, as opposed to machine language, has resulted in a magnification of each programmer's effort. The resulting programs are easier to understand, thus greatly decreasing the startup time when a new programmer is assigned to work with a program written by another. They are also easier to revise — or to replace entirely, for that matter — if a module must be redesigned: a several-pages-long EPL program is still less intractable than a single-page-long EPLBSA (assembly language) program. On the other hand, system efficiency becomes highly dependent upon the EPL compiler's relative efficiency. Far from being a disadvantage, this means that attention can be focused on code optimization in the compiler, so that a single improvement here "automatically" improves all modules of the system.

Originally developed as an interim device, pending delivery of a PL/I compiler produced for the GE 645 by an outside contractor, EPL became the compiler for the present Multics effort when the PL/I delivery did not materialize. Because of EPL's position of central importance to both development and operation of the Multics system, considerable effort has been directed at minor expansions of the language and at upgrading the compilation process. Late in the year, the responsibility for EPL development and maintenance was transferred from the original development team at Bell Laboratories to a permanent support group at GE in Cambridge.

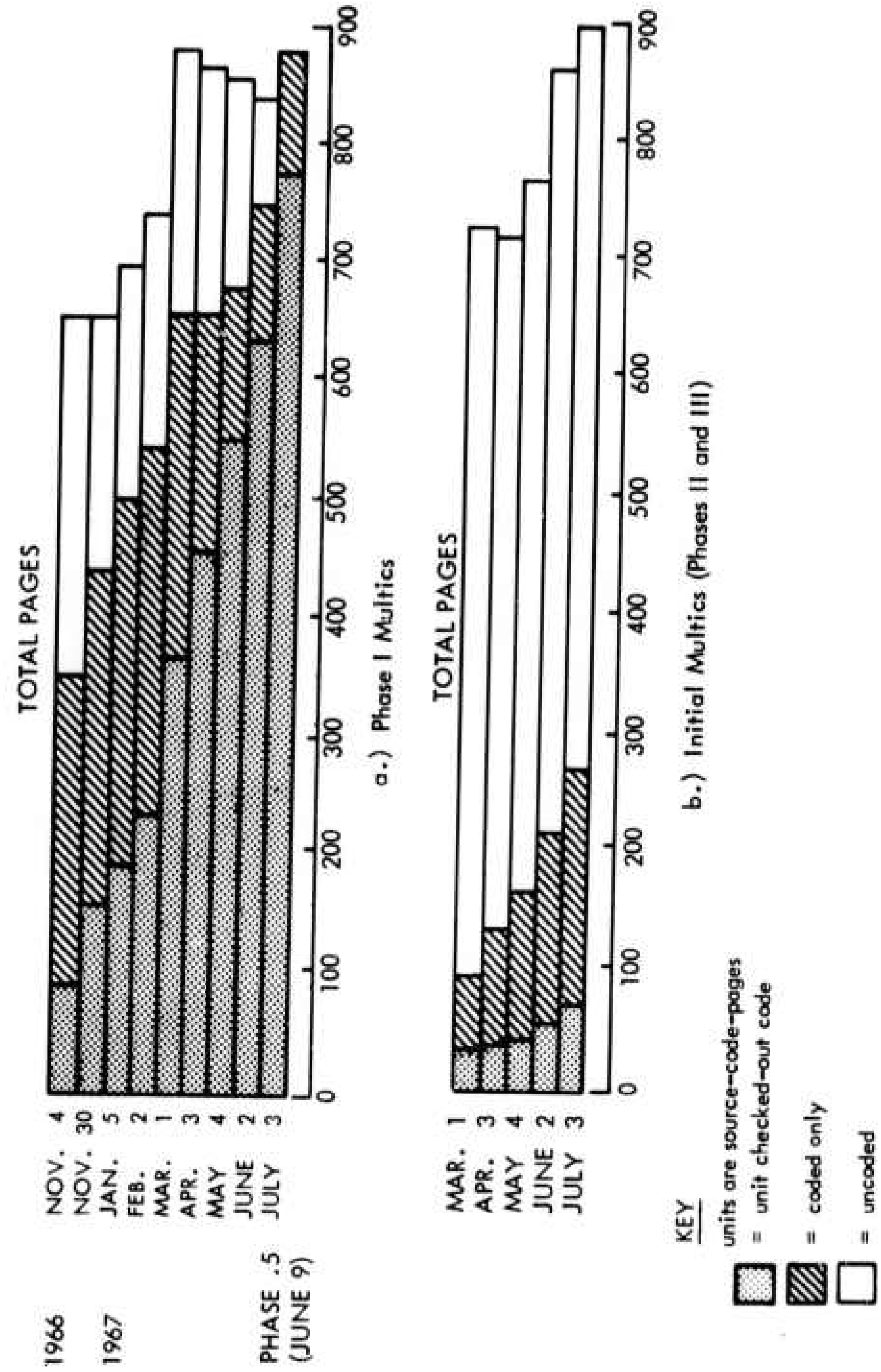


Figure 7. The Growth of Multics Software

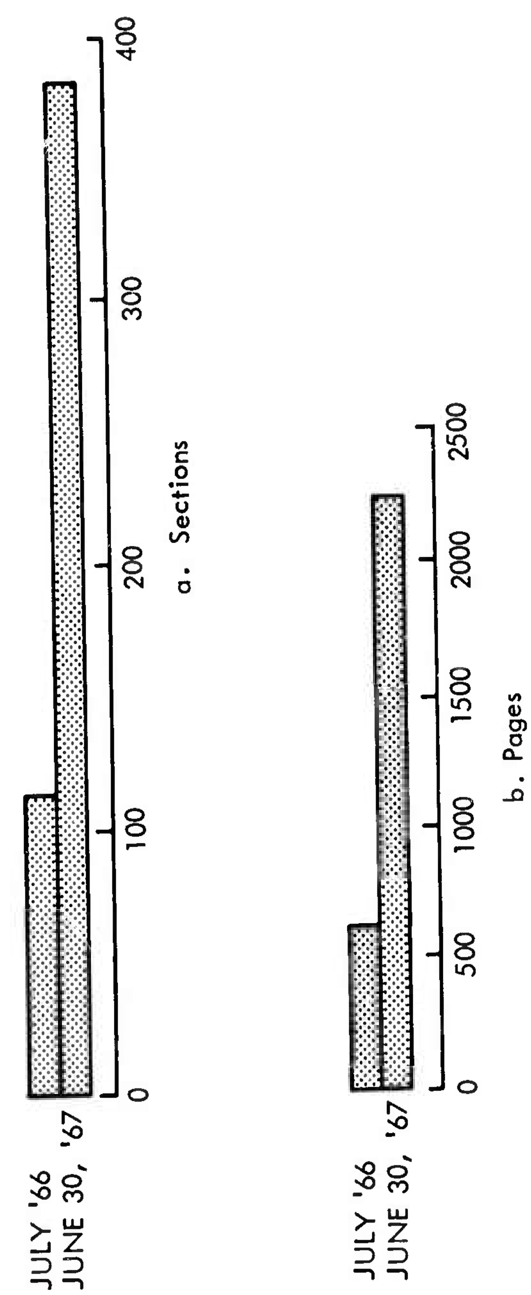


Figure 8. Growth of the Multics System Programmers' Manual

Dependence of the project on the testing and debugging system has been mentioned. During the reporting period, a number of improvements were made in this system. They include:

1. Installation of a drum memory to house the testing and debugging system, formerly maintained on the lower-speed disc memory;
2. A series of improvements to GECOS, the basic monitor for the GE 635, made by General Electric personnel;
3. Delivery of the GE 645 hardware, with its monitor K2-GECOS, which allows direct use of the GE 645 instruction set;
4. Substantial improvements to the subprogram loader in the testing and debugging system; and
5. Installation of an abbreviated memory-dumping procedure. (This significantly reduced the amount of redundant information returned from a run on the testing and debugging system, thereby both decreasing input/output time and easing CTSS file memory requirements.)

D. MULTICS IMPLEMENTATION -- BY AREA

The project has been divided into several functional sub-project groups, each consisting of staff members from one or more of the three cooperating organizations, and responsible for a specific aspect of the overall effort.

The groups are currently working in the following areas:

Command System

File System

Language Development

Central Supervisor

Hard-core I/O

I/O

General System

The Command System group is responsible for creating an interface between a casual user and the Multics system. During the past twelve months the group succeeded in

1. making the command language interpreter (the Shell) operational;
2. completing work on the RENAME, DELETE, LINK, and LIST commands, making these operational; and

3. writing a context editor, EDIT, and putting it in working order.

The File System group, responsible for the development, debugging, and integration of the Phase I File System, including the necessary system initialization for the File System, succeeded in

1. demonstrating the workability of paging, virtual memory, and the directory hierarchy; and
2. coding the self-initialization portion of the file system.

The Language Development group, primarily a GE activity, supplies Multics language processors. The responsibility for continued maintenance of EPL was assumed by this group during the Spring.

The original EPL compiler was designed to operate under CTSS on the IBM 7094. Using the testing and debugging system, the resultant code was then assembled with the EPLBSA assembler on the GE computer prior to execution. It is planned to make the EPL compiler available . the GE 645 due shortly after the end of the reporting period.

Several improvements in the compiler were initiated during the past year:

1. Continued debugging of the EPL compiler made it more reliable;
2. Improved diagnostics and language capabilities made it more flexible and of more general use;
3. Development work was started on an additional pass for the purpose of code optimization — as a result of this work, the compiler will produce significantly more efficient object programs; and
4. Conversion of the EPL compiler for direct use on the GE 645 was begun — this will improve the speed of service by shifting some of the load from CTSS to the GE 645.

In addition to this task, work is being done on several other language projects:

1. The design and implementation of a new compiler which will ultimately replace the EPL compiler;
2. FL/I (Function Language I), a macro assembler; and
3. FORTRAN IV.

The Central Supervisor group, in addition to its primary responsibility for the Traffic Controller area, during the past year worked on the design of several important parts of the supervisor system:

1. System initialization,
2. Interprocess communication, and
3. User control subsystem (the Overseer).

The Traffic Controller is a primary aspect of Multics. Its design is given in MAC-TR-30, Traffic Control in a Multiplexed Computer System, July 1966. (See Appendix D.) During the reporting period, coding and checkout proceeded on the modules of the Traffic Controller.

The Hard-core I/O group was responsible for developing a workable software interface to the General Input/Output Controller (GIOC), and has succeeded in this goal with its development of a working GIOC Interface Module (GIM).

The I/O group was responsible for developing a workable interface between the system innards and a console user, and succeeded in developing the capability for the system to communicate with a number of input/output devices.

The General System group is responsible for a number of support activities, including documentation, the testing and debugging system, and computer resource allocation. The group is also responsible for system standards and services. During the year, the group's activities included:

1. Design and initial coding of the Protection Mechanism, and the associated Signal and Abnormal Return facilities;
2. Coding and check-out of the Linker and linkage maintenance routines;
3. Design and initial coding of the Binder and post-binder; (The function of the Binder is to combine separately-compiled segments into single segments after the separate segments have been checked out; this ability will be of immense value in a working Multics system — and, indeed, in later phases of development — because of the decreased overhead it affords although it does of course decrease the amount of flexibility of the bound segments with regard to changing component segments.)
4. Extensions to the EPLBSA assembler, coding, and check-out;
5. Maintenance of the Segment Inventory;
6. Maintenance of the Multics System-Programmers' Manual (MSPM); and

7. Development of an on-line administrative system for CTSS resource allocation for the Multics staff.

The Multics System-Programmers' Manual, whose contents may be viewed as reflecting the relative size and progress of the project, experienced substantial growth during the year. In July of 1966, the manual had 111 sections covering 609 pages. At the end of June, 1967, the manual had 373 sections covering 2258 pages. During the year, 262 new sections were added to the manual, while 66 more underwent at least one revision. (See also Figure 8.)

E. SPECIFICATION OF BENCHMARKS

Several key interim points, or "benchmarks", in the development of Multics have been specified. Although these were established on a serial time scale, effort is currently directed at them in parallel, with greater emphasis on achieving the earlier benchmarks.

Phase .5 consists of the Phase I File System running under the testing and debugging system, and available to a user which is either the Command System or the test and diagnostic programs. Communication during running of test and diagnostic programs requires a GE 646 on-line typewriter. The purposes of Phase .5 are to gain confidence in use of a checked-out File System, checkout of the Command/File System interface, testing of hardware interaction of the processor and drum, and checkout of the paging mechanism. Phase .5 was achieved on June 9, 1967.

Phase I is the basic Multics system framework operating with a single process and a single console. The single process is a "tied-together" version of the overseer, working, and device-manager processes which are normally associated with each typewriter. The principal reasons for Phase I are twofold:

1. It allows exhaustive testing of the File System hierarchy, and

2. All experiments are reproducible.

In addition, the integration of the general scheme of commands and I/O is begun. The Multics system is self-initializing from a reel of magnetic tape.

Phase II has multiple typewriters able to asynchronously login, operate commands, and logout. The system is still vulnerable to mistreatment. Phase II allows the exhaustive testing of:

1. the basic user-user and user-system protection machinery,

2. the traffic controller, and

3. the vital machinery of interprocess communication.

In addition, commands continue to be added and I/O modules continue to be added and improved.

Phase III has the system in daily operation and able to accommodate arbitrary, self-sufficient, uncomplaining, constructive system programmers. The system will for the first time be available for sustained individual work, although it will have innumerable bugs and omissions to be maneuvered around. System programmers will, for the first time, be able to accummulate their own files of programs and work in parallel on all parts of the system. Major accomplishments required for Phase III are:

1. the beginning of an effective file backup system to ensure preservation of project work,

2. the beginning of an individual file retrieval system to allow repair of localized mishaps,

3. the ability to read and write magnetic tapes, and

4. the addition of a disc or Race files, as well as multilevel storage management to allow a large volume of file storage.

Phase IV resembles Phase III except that the rest of the Project MAC population become users (i.e., the shakedown test population), and the number of bugs and difficulties are reduced to the point where the burden of education and complaints arising from the use of the system is tolerable.

A recent change in the definition of the phases of development created the notions of "Initial Multics" and "Prototype Multics". Initial Multics essentially replaces Phases II and III; Prototype Multics replaces Phase IV.

The purpose of Initial Multics is to provide a proper subset of full Multics which will be available during 1968, and which can demonstrate the unique and principal concepts of Multics integrated into a single time-sharing system. Initial Multics is to be capable of maintaining and extending itself dynamically with the exceptions of the EPL and EPLBSA language processors.

Any required maintenance of these processors will be accomplished using the K2-GECOS monitor and will be performed during normal Initial Multics shutdown periods. The Initial Multics system is to provide simultaneous service to a number of remote console users. Use of the system is to be restricted to Multics system programmers and selected uncomplaining subsystem designers. Performance of the system in terms of space or time efficiency or the ability to handle a large number of user terminals is not of principal concern.

Prototype Multics incorporates all remaining work not included in Initial Multics, and will represent the development of a system capable of use by the casual and, of perhaps greater importance, less friendly user.

Development of Multics will not, of course, be "completed". Like CTSS, Multics will continuously evolve as more users, subsystem writers, and system programmers use it and contribute some of their individual efforts to the repertoire of the Multics system capabilities.

F. HARDWARE

During the reporting period, the interim GE 635 was replaced by a dual-processor GE 645 at Technology Square in Cambridge. The GE 635 was retained while the 645 was being phased in, and then was removed in the Fall of 1966. In July of 1966, a one-processor GE 645 configuration arrived; by July of 1967, the full dual-processor configuration was on site. This equipment has been placed in daily operation under supervision of both MAC and GE personnel. Several increases in the memory capacity of the computer have been made during this period. These increases have not only greatly facilitated the software debugging effort, but are also necessary to support a running Multics system. The table below lists the GE 645 hardware on site at the beginning and end of the reporting period.

The current on-site hardware is prototype hardware. Plans are being made to replace it with production versions of the same devices, sometime in 1968.

GE 645 On-Site Hardware

As of July 22, 1966

Quantity	Item
1	645 CPU
—	65K Memory
1	Power Supply
10	Tapes
2	DS-20 Disc Files
1	Input/Output Controller
1	Console
1	Peripheral Switch
1	PRT 202 Printer
1	CRZ 200 Card Reader
1	CPZ 200 Card Punch

GE 645 On-Site Hardware (continued)

As of July, 1967

Quantity	Item
2	645 CPU
—	256K Memory
2	GIOC
1	4-million-word, high-performance Drum
8	Tape Drives
1	Peripheral Switch
2	PRT 202 Printer
2	CRZ 200 Card Reader
2	CPZ 200 Card Punch
1	Console
2	DS-20 Disc Files
2	System Clock
1	Drum

A Data Storage Structure for Multiplexed Information Systems - Arthur A. Bushkin

The overall research objective is the specification of a data management facility for a multiplexed computer system. This system design is the intended area of the author's doctoral thesis, and a preliminary investigation of file structures for on-line systems constituted the author's master's thesis.

The concept of a data base as a formal, graph theoretic characterization of an information file was presented for any information file with nonrecursively defined information types. The logical structure of the data base is then used to map the set of data items associated with the data base into one of its linear subsets, called the data stream. These formal ideas led to a technique for totally specifying the logical interrelationships between the data items. Extensions to the graph theoretic approach leading towards the design of an internal data storage structure for an on-line, multiplexed information storage and retrieval system were then indicated.

A Communications Bus Machine - Daniel J. Edwards

One method of bringing more computing power to bear on discrete mathematics problems, which require from 30 to 300 hours of main-frame processing time, would be to use several computers in parallel. Here we mean true parallel processing (i.e., several computers working on pieces of the same program), rather than multiprogramming which implies several connected computers working on more or less separate problems. The chief drawback to true parallel processing is the method in which several computers working on the same problem communicate with each other. The proposed parallel computer consists of one medium-scale general-purpose computer tied to many small-scale general-purpose computers via a communications bus. We call the entire collection of computers and the communications bus a communications bus machine.

The medium-scale computer supervises the actions of the small computers and also handles all the I/O for the bus machine. The small-scale computers, each of which runs asynchronously with respect to the other computers, actually process the data. The communications bus allows any machine to transmit a word of data to any other machine in a length of time equal to the small computer memory cycle. While such a machine may never actually be built, the discussion of a specific machine design may serve as a point of departure for those who wish to pursue the bus communications philosophy further. The medium-scale general-purpose supervisor computer proposed for the bus machine would be a Digital Equipment Corporation PDP-10/20, and there would be ten small-scale general-purpose computers, each being a Computer Controls Corporation DDP-516.

The communications bus is simply a set of 16 parallel data lines, 6 parallel address lines, and a number of common control lines. The bus control logic is distributed in the communications bus interfaces in the computer. For control purposes, all of the computers may be thought of as connected in a ring. When bus control arrives at a particular computer, it places data on the bus if it has any ready. If not, it passes bus control to its neighbor.

The following observations can be made about the bus machine. Each machine on the communications bus can react to four different bus addresses. The bus has been designed so that two or more machines can react to the same bus address and thus receive the same data. In addition, each small machine can signal the supervisor and vice versa. In automatic-shift mode, the bus runs asynchronously with respect to all the computers. Since it takes four to six memory cycles (including priority interrupt) to unload the bus data buffers, an overall bus operating speed of about one work transfer per small machine cycle time appears to be sufficiently fast to keep the bus machine running at full efficiency.

Programming the bus machine, we find each small machine has a protected portion of memory containing a loader which is executed when the supervisor interrupts the small machine. Using this loader, the supervisor can load programs or data into the small machine. The supervisor can also force the small machine to execute given instructions via the loader.

Each small machine should have its own multiprogramming monitor programmed along the lines suggested by Hornbuckle.* This would permit a small machine to handle several small tasks efficiently, depending on which bus address it was reacting to.

On the larger scale, a modified ALGOL translator should be available to write programs for both the small machines and the supervisor computer. The translator should run on the supervisor computer and at least one other large-scale general-purpose computer. In addition, a standard package of assemblers, loaders and debugging aids should be available.

The last piece of recommended software would be simulators for both the small machines and the entire bus machine written for another large-scale computer. This facility would aid programmers in checking out pieces of bus machine programs without tying up the bus machine itself.

The communications bus organization for a parallel computing machine would provide an interesting vehicle for research in parallel computing techniques. The total hardware cost involved is on the order of $500K. The bus machine at its best should allow certain problems in discrete mathematics to be solved 5 to 10 times faster than possible on present-day hardware. Furthermore, the bus machine programmer would have available modern programming techniques for writing programs. Since most of the hardware in the proposed machine is commercially available today, the expected lead time to get this particular bus machine on the air is from 12 to 18 months.

Over and above the specific machine proposed here, the communications bus philosophy for making sensible use of parallel computing machines deserves further consideration towards the goal of achieving high-speed computing via parallel processing.

---

*Hornbuckle, G. D., "A Multiprogramming Monitor For Small Machines", Communications of the ACM, vol. 10, no. 5, May 1967, pp. 273-278

BLANK PAGE

ELECTRONIC SYSTEMS LABORATORY

Academic Staff

M. L. Dertouzos
L. A. Gould
F. J. Reintjes
A. K. Susskind
K. Zander

Non-Academic Research Staff

A. K. Bhushan
R. J. Bigelow
T. B. Cheek
R. W. Cornew
C. G. Feldmann
E. G. Guttmann
D. R. Haring
F. B. Hills
J. Katznelson
P. T. Ladd
P. Z. Marmarelis
R. C. Nelson
R. B. Polansky
D. T. Ross
R. H. Stotz
W. D. Stratton
J. R. Sussman
D. E. Thornhill
A. Vezza
J. F. Walsh
J. E. Ward
T. S. Weston
B. L. Wolman

Research Assistants and Other Students

L. J. Bernhardt
F. T. Blount
R. H. Bryan
L. O. Craft
R. J. Diephuis
M. Edelberg
D. S. Evans
H. L. Graham
I. Greif
M. M. Hammer
W. Hutchinson
W. M. Inglis
P. G. Jessel
M. E. Kaliski
R. B. Lapin
H. D. Levin
B. K. Levitt
K. P. Polzen
R. L. Randall
J. E. Rodriguez
J. J. Sierakowski
J. R. Stinger
J. R. Stone
E. J. Tejara-Rodriguez
C. W. Therrien
E. R. Vassar
Y. D. Willems

Guests

S. I. Ackley - System Development Corp.
D. Barovich - I. B. M. Corporation
A. J. Berger - Boeing Company
W. L. Johnson - Ford Motor Company
J. H. Jones - McDonnell Aircraft Corp.
F. Luccio - Olivetti Underwood Corp.
R. S. Lynn - North American Aviation
R. A. Meyer - I. B. M. Corporation
A. K. Mills - Dow Chemical Co.
A. T. Nagai - Boeing Company
J. H. Porter - Chevron Research Co.
I. Wenger - Raytheon Company
R. B. Wise - ITT Research Institute
S. Zurnaciyan - North American Aviation

Introduction - John E. Ward

The Project MAC time-sharing system continues to stimulate the research activities of a substantial number of faculty, staff, visiting staff, and graduate students of the Electronic Systems Laboratory. In addition, a number of other graduate and undergraduate students have found opportunities for thesis research in connection with MAC/ESL activities.

During the past year, MAC-related research in ESL ranged over a broad spectrum, including display system technology, programming systems and languages for computer-aided design, computer-aided electrical network design, planning for a future M.I.T. network of remote-access terminals, and library information retrieval (Project Intrex). These topics are discussed in the following sections, except for Project Intrex which is reported elsewhere in this volume.

Part of the display research in the Electronic Systems Laboratory is directly supported by ARPA through Project MAC: other MAC-related research in the Electronic Systems Laboratory is supported by a number of agencies, including: Fabrication Branch, Air Forces Materials Laboratory, WPAFB: Avionics Laboratory, USAF, WPAFB: and the National Aeronautics and Space Administration.

Display Systems Research - John E. Ward and Robert H. Stotz

For several years, the ESL Display Group has been conducting research and development in the field of computer-driven CRT displays and related equipment for Project MAC and the ESL Computer-Aided Design Project. As previously reported, the group has developed a number of devices, including the ESL Display Console, a scan-conversion system to convert computer displays to TV format, and a prototype low-cost remote graphic terminal for time-shared computer systems. Progress during the past year included improving these systems, and planning future displays for the Multics system.

A. ESL DISPLAY CONSOLE

As described in previous annual reports, the ESL Console was built in 1963 and has been operating on the Project MAC time-shared 7094 computer system since early 1964. Although originally designed for use in computer-aided design, the console has proven to be a versatile input/output tool in many other areas of study, such as speech analysis, molecular biology, naval architecture, and civil engineering. The only console modification during the reporting period was the installation of dual-deflection CTR's, which has permitted a four-fold increase in character writing speed. The major new innovation was the installation of a buffer computer.

During the past three years, the ESL Console has operated without a buffer memory, depending on the 7094 supervisor program (A-core) for display data storage and for real-time display manipulations. To enhance the capability of the ESL Console and to lighten its load on the CTSS main computer, a small general-purpose computer (Digital Equipment Corporation PDP-7) was acquired in February, 1966 to drive the display. Interface equipment (described in MAC Progress Report III) was built to couple the PDP-7 to both the display and the 7094. The PDP-7 installation at Project MAC is shown in Figure 9, and the ESL-constructed interface (installed in the upper right-hand bay of the PDP-7) is shown in Figure 10.

Programming to permit buffered operation is in the final stage of debugging. To aid in the programming work, the 7094 BEFAP assembler was modified to handle PDP-7 code. The system thus allows a user to write and assemble PDP-7 programs on CTSS using all its powerful editing tools, and then load the PDP-7 directly using a modified version of AED's relocatable loader. Various PDP-7 utility programs (DDT, etc.) are also resident in the CTSS file system. The buffered system should be operational this summer.

During this past year a duplicate of the ESL console, shown in Figure 11, was built to our specifications by Digital Equipment Corporation, and was checked out and installed at the M.I.T. Computation Center.

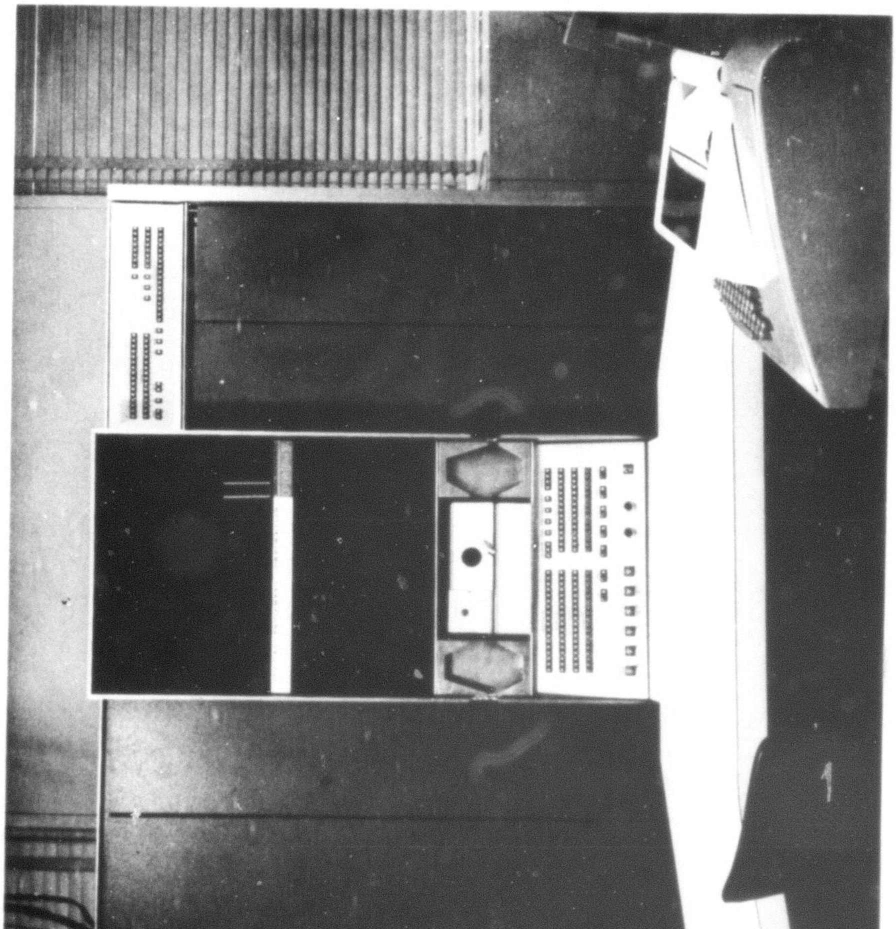


Figure 9. PDP-7 Display Buffer Computer Installed at Project MAC

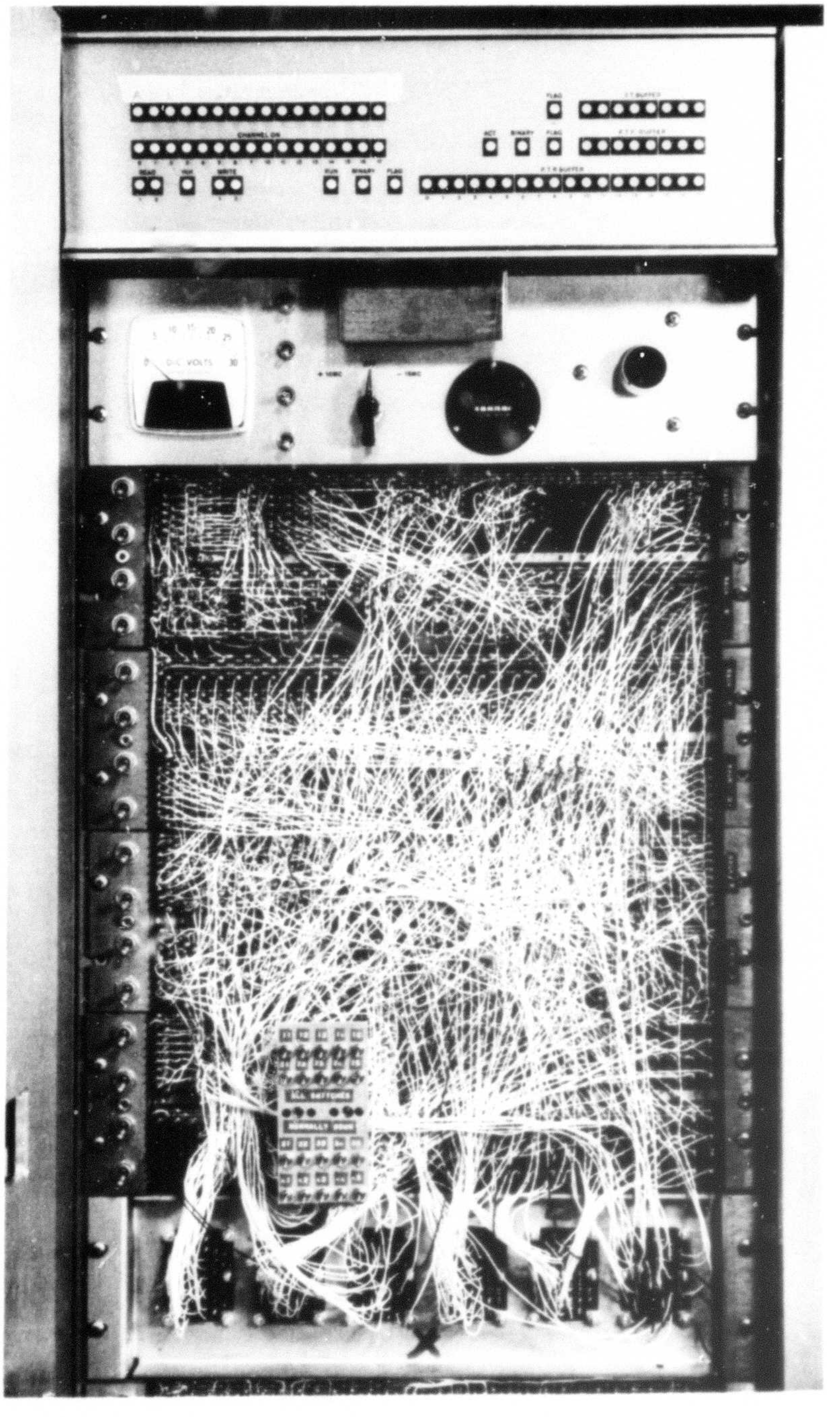


Figure 10. 7094/PDP-7/ESL Console Interface Installed in PDP-7

Figure 11. Number 2 ESL Display Console at M.I.T. Computation Center

Since the Center's version of CTSS differs only silightly from that at Project MAC, the Console became operational without difficulty, and was made available to users in December, 1966.

B. INVESTIGATION OF DDA ROTATION MATRIX

A unique feature of the ESL Display Console is its hardware rotation matrix, which allows real-time rotation of three-dimensional complex figures with a minimum of load on the controlling computer. This equipment was implemented using binary rate multipliers which, although inexpensive, produce round-off errors that impair display quality. A digital differential analyser (DDA) rotation matrix using multi-level threshold logic was first investigated by E. Guttmann in a master's thesis research in 1965. Subsequent work leading to an improved threshold adder cell was presented in last year's annual report. This past year, work has continued on developing a complete DDA rotation matrix to be installed in a DEC 340 Display at the M.I.T. Science Teaching Center. At the time of this writing, circuit design, layout, and packaging have been completed and a four-stage adder is being checked out.

C. LOW-COST DATAPHONE-DRIVEN GRAPHIC DISPLAY

Last year's annual report discussed the need for low-cost graphic terminals for computer time-sharing systems and our efforts to build such a unit, based on a direct-view storage-tube approach. This past year we have expanded upon the breadboard model (ARDS-I) previously reported and built a full prototype console (ARDS-II) which contains a full-screen vector generator and a character generator for the full ASCII symbol set (95 printable symbols). ARDS stands for advanced remote display station.

The principal thrust of this research has been to make the cost of the console as low as possible. For this reason, integrated circuits have been used throughout the digital portion of the logic. Also, an integrated diode array (6,144 diodes on a 1/2 inch sq. chip) was chosen for the read-only memory of the character generator because of its low cost relative to other memory techniques. A new technique for converting the digital line information into analog voltages has been developed which significantly reduces the cost of this part of the equipment, and a patent application has been filed on this technique.

A block diagram of the output portions of ARDS-II is shown in Figure 12. All communication is in ASCII format, and the unit responds according to one of four modes established by assigned control characters: symbol, setpoint, short vector, or long vector. In symbol mode, each printable ASCII character is displayed in typewriter-like format. In setpoint mode, four ASCII characters provide ± 10-bit binary-coded components for X and for Y. Long vectors require four characters for ± 10-bit ΔX and ΔY components. Short vectors use two characters to provide ± 5-bit ΔX and ΔY components.

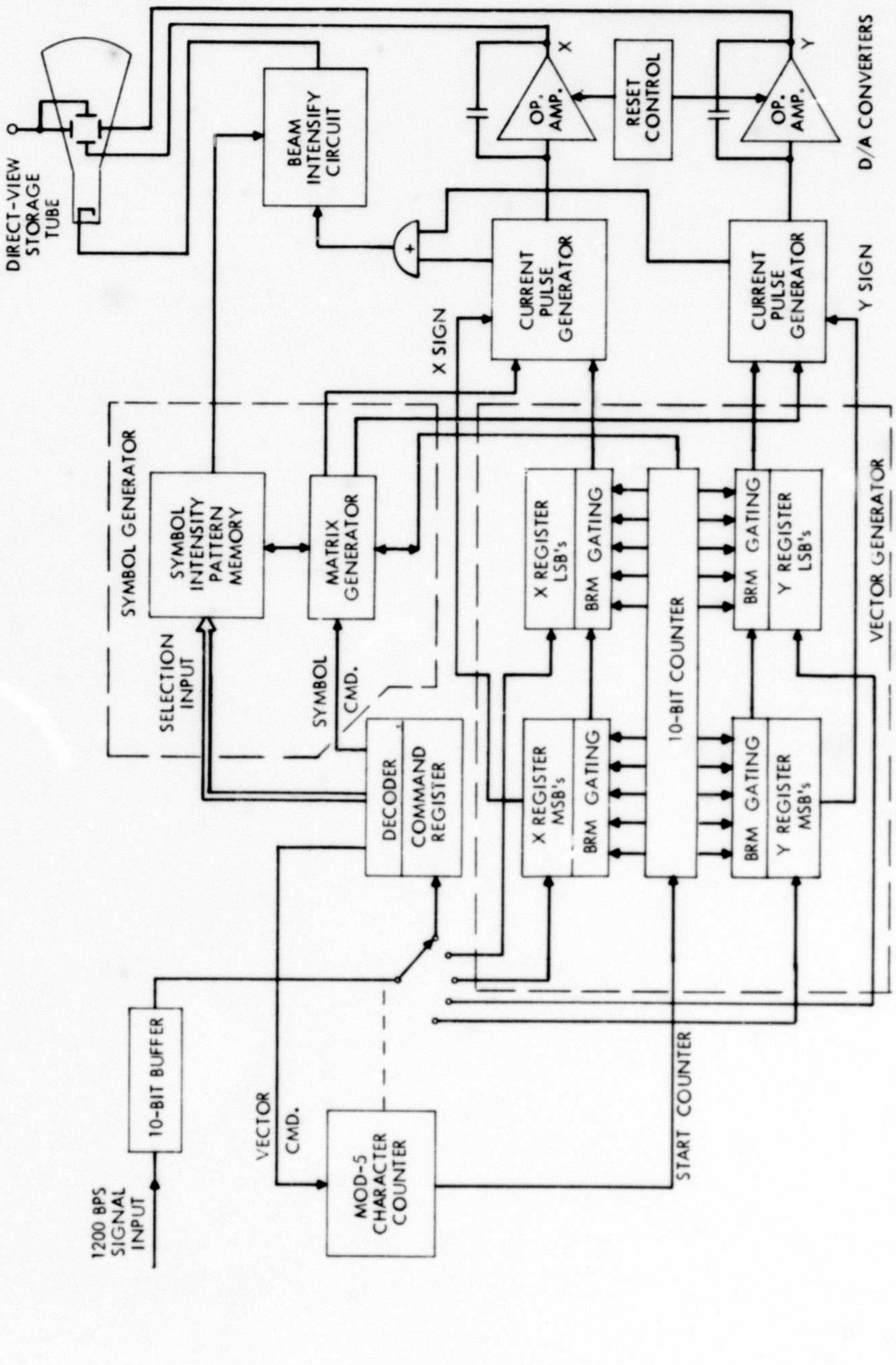


Figure 12. Block Diagram of ARDS-II (Output Portions Only)

The low-cost display is now reasonably complete as an output device, and is in operation on the 7094 CTSS via a 1200-baud line from the 7750 interface computer. Because the 7094 cannot accept data input over this 1200-baud line, a Model 37 Teletype with a separate 150-baud line to the 7750 is presently serving as our input means. Figures 13 through 15 show the present ARDS-II terminal, the diode array, and the complete character generator. The remaining hardware additions concern only input devices, and these should be complete within a few months. A keyboard is on order, and several graphic input units have been demonstrated, as related in the next section.

The next vital step will be to develop software to make this display operate in the system like a teletype. However, to truly test the effectiveness of the unit as a terminal we will require a storage tube larger than the present five-inch tube, and with better resolution. We expect to acquire a prototype of such a large-screen high-resolution tube this summer which will enable us to display up to 4000 characters. (See Stotz, Appendix C.)

D. RELATED DISPLAY TECHNOLOGY

A graphical input device called "The Mouse" (by its inventors at Stanford Research Institute) was built for use with the low-cost display. The mouse is a hand-held device which rests on two wheels that are mounted at right angles. These wheels resolve the motion of the mouse across a desk top or other surface into x and y components, and drive shafts of low-torque potentiometers, whose voltages then represent the x, y position of the mouse. This is being tested as a possible inexpensive graphic input device for the remote terminal.

Other work being conducted to extend the usefulness of the remote terminal includes a study of possible remote hard-copy generation techniques. At present a dry-silver photographic paper developed by the 3M Company looks very promising.

Two undergraduate theses completed this past spring were related to the low-cost terminal. In the first, an inexpensive writing stylus was built by F. Blount. This device uses a resistive paper (Teledeltos) as the tablet surface, and a common ball-point pen as the stylus. A d-c field is placed across the paper and the pen acts like the arm of a potentiometer moving across the paper, its voltage corresponding to distance from an edge. The field is alternately applied to the x and y directions of the paper at one-millisecond intervals, so each component is sampled 500 times per second. The total cost for parts of this device is well under \$100. Linearity of about 1% was measured. (See Blount, Appendix B.)

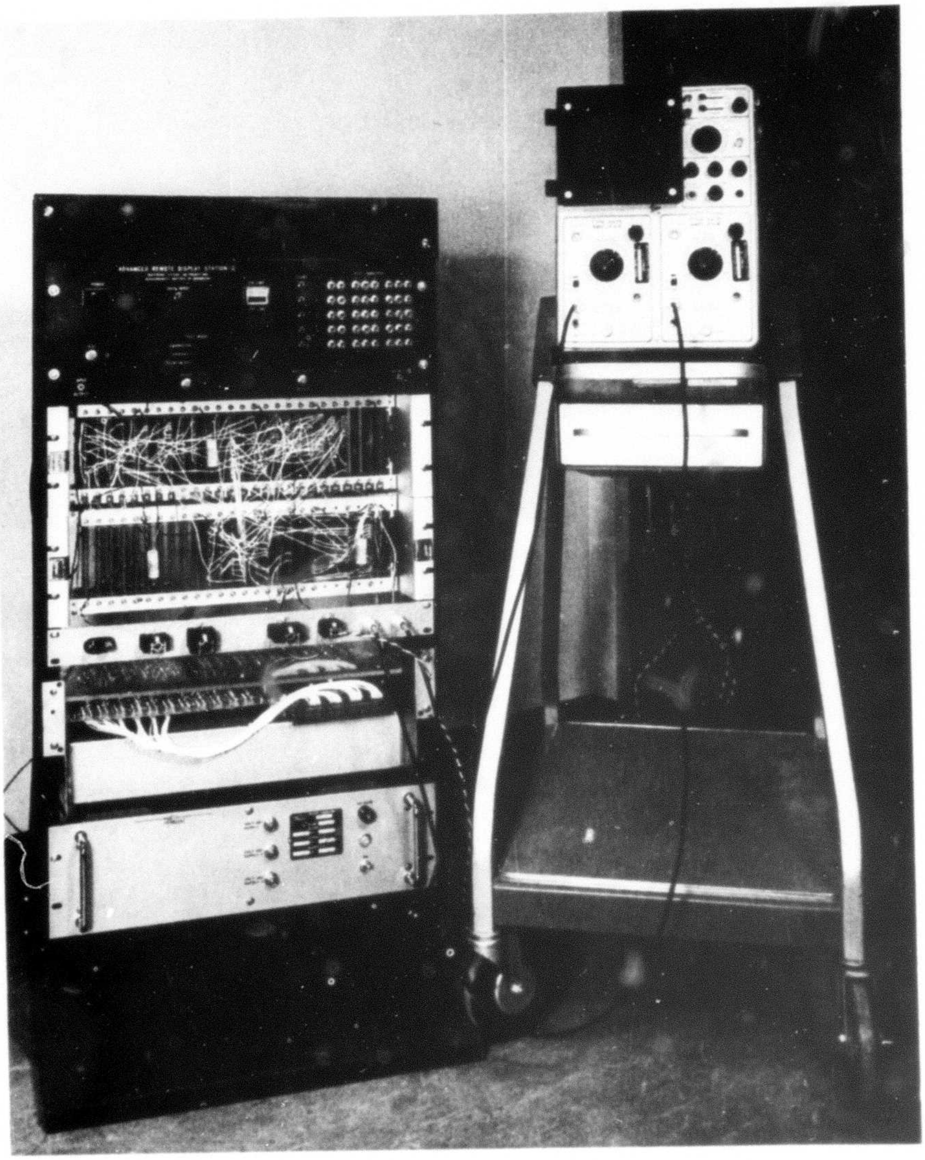


Figure 13. ARDS-II with Tektronix 564 Storage Oscilloscope

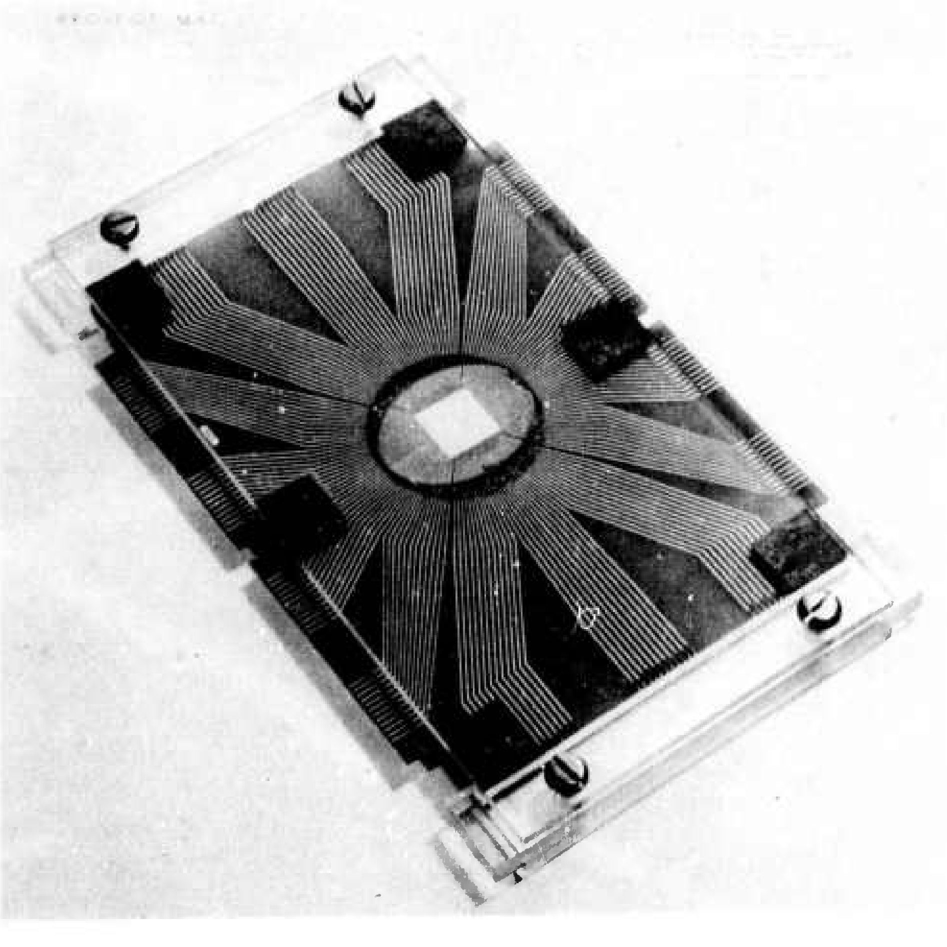


Figure 14. Autonetics 96 x 70 Diode Matrix

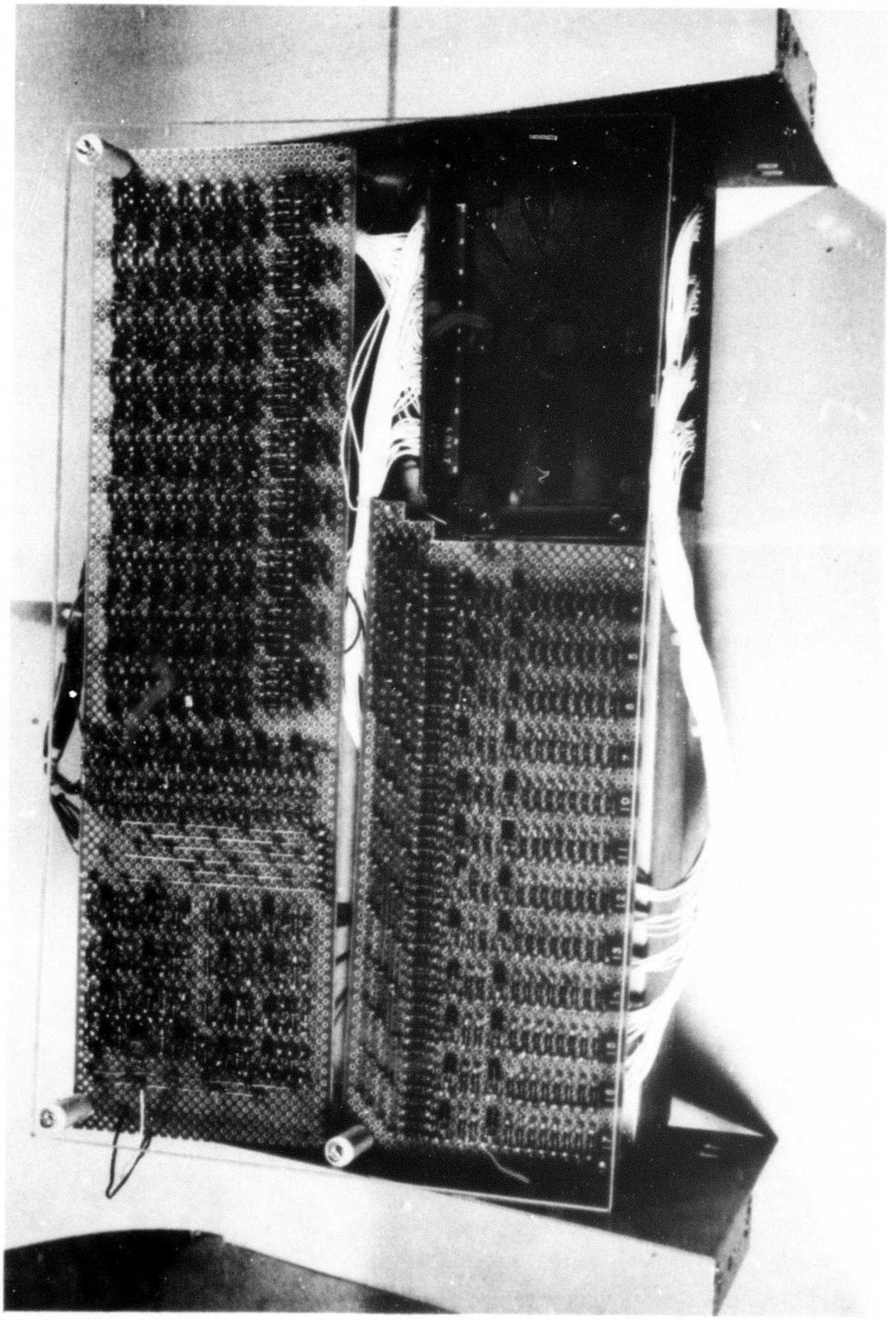


Figure 15. Read-only Memory for Character Generator with Diode Matrix, Selection Logic, and Sense Amplifiers

The second thesis, by E. Vassar, dealt with a telephone-line communications problem of the ARDS-II terminal. At present, one must rent a Bell 202C data modem ($45 per month) and have it installed on a special telephone line. An audio-coupled data set that would allow a terminal to be used from any standard telephone would be of great advantage. Couplers now on the market operate only up to 300-baud, however. The attempt in this thesis project was to build the receiver section of an audio-coupled or magnetic-coupled data set that could receive digital data at 1200 bits/second. Although the project was not completed, enough experience was gained so that we feel confident we can receive at this rate. (See Vassar, Appendix B.)

Another area of research of the ESL Display Group has been an effort to design a wideband communications network for coupling satellite computers, with displays, to the Project MAC GE-645 computer system. The communications format and protocol for such a net has been examined and a proposal for the configuration and equipment is being formulated. The concept of this network is that it will eventually tie together computers from all over M.I.T. and possibly outside the campus.

The last area of study of the display group is in new techniques for high-performance display systems. This includes vector generation techniques, hardware rotation capabilities, and study of the proper coupling between a display control unit and a small general-purpose computer. As part of an effort to determine the feasibility of speeding up the digital display generation techniques used in the ESL Console, a 10-Mhz binary rate multiplier was designed as an undergraduate thesis by L. Bernhardt. (See Bernhardt, Appendix B.) Also completed during the year was a study of an improved beam pen, carried out as Master's Thesis by B. K. Levitt. (See Levitt, Appendix B.)

Computer-Aided Design Project - Douglas T. Ross

The M.I.T. Computer-Aided Design Project, sponsored by the U.S. Air Force under Contract F 33615-67-C-1530, is a research program on the application of modern data-processing concepts and techniques to the design of mechanical parts, as an extension of automatic programming (APT) systems for numerically-controlled machine tools.

The major emphasis continued on the AED (Automated Engineering Design) family of programming systems including: the AED-1 System, whose domain is general programming, compiling, and operating of programs on essentially any large-scale computer; the AEDJR System, which builds a parsing processor; and the CADET System (Computer-Aided Design Experimental Translator), aimed at a generalized approach to computer-aided design applications. Progress in several technical areas

included 1) a major revision of the RWORD Package, which builds items from the input character stream, 2) incorporation into the general problem-solving scheme of a revised AEDJR which enlarges the present realm of possible AED applications, 3) major improvements in preprocessing, and in second pass, 4) introduction of a "Features Feature," which permits selecting only those facilities relevant to a particular problem situation, 5) improved integrated packages, 6) initial phases of bootstrapping AED to the Univac 1108 and IBM 360 computers, 7) a MAD-to-AED translation program, and 8) completion of AEDNET, which simulates nonlinear electronic circuits. The AED-O System was used in all of this work.

To assist in the dissemination of research results, the Project continued the AED Cooperative Program in which programmers from industry join in the research effort at M.I.T. Increased activity and interest is reported in these cooperative phases of Project work, including the First and Second AED Technical Meetings.

A. MAN-MACHINE PROBLEM SOLVING

The objective of the Project is to develop a language-independent and machine-independent generalized translator system to enable man-machine cooperation in problem-solving. The requirements for such a translator do not vary with time, for solving any one problem imposes a variety of requirements which must be satisfied in solving any other problem.

These problem-solving requirements have been revealed in increasing depth during this research effort and are reflected in the design of the generalized translator scheme being used for the AED-1 processor, shown in simplified form in Figure 16. The major phases are shown within the blocks, and the data output of each phase is identified beneath the arrows connecting the blocks. To compile a programming language, such as AED-O, the input string of characters must be converted into a string of multi-character items — the words, syllables, and punctuation marks of the program. A first pass is made, parsing these items into a tree structure, which shows how the items are grouped to form phrases, and how the meanings of these phrases are to be combined to build up meaningful, complete statements. A second pass is made, translating these meanings and producing a delayed-merge structure, which is finally converted into an output character stream in symbolic assembly language. A standard assembler then yields a machine-coded program.

The modular nature of this translator system permits each of the major phases to be improved without disturbing other portions of the general scheme. It also serves (with some modification) as a generalized scheme for nonprogramming languages, including graphical language.

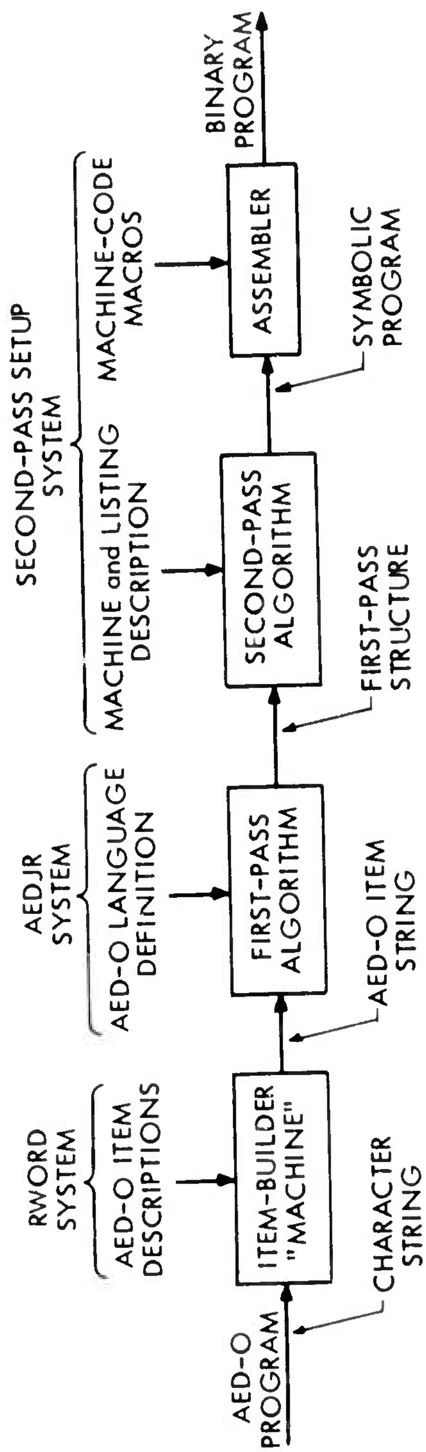


Figure 16. General Problem-Solving Scheme

Figure 16 thus can serve as a map of the general translation scheme of the Project, showing the relationship between any one component area and the total translation process.

B. THE AED-1 PROCESSOR

The AED-1 processor is being designed to handle the AED-O language initially; later language expansions will result in an AED-1 language. All of the major programs for the front end of the AED-1 processor shaped up strongly during the year, as did many parts of AED-1 Second Pass. The overall second-pass problem, described in the next section, seems to require more study, however, and to get started bootstrapping AED to other computers we decided to base the initial second pass for AED-1 on the AED-0 second pass. The strategy is to make the AED processor (running on the 7094) produce assembly language code for the target machine, which can then be run through the normal assembler of the target machine to produce a binary object program.

Initial phases of bootstrapping to the IBM System 360 are being done in collaboration with the IBM Cambridge Scientific Center, and some AED packages are already running on the 360. We hope to have a complete 360 AED by early Fall. Bootstrapping to the Univac 1108 is being accomplished largely by United Aircraft Corporation, and good progress has been reported. The Project plans to initiate work directed toward the GE 645 as soon as Multics System schedules permit.

Preprocessing: Considerable progress has been made programming the preprocessing phase for AED-1. During preprocessing (which precedes parsing), all declarations of symbols are performed in conjunction with the construction of the block-structured symbol table. Macro definition and expansion also takes place at this time. All of these operations are controlled by making a subset of the AED-0 language vocabulary active during preprocessing, and using the full first-pass parsing to recognize context and control the actions of the various routines involved. Output from preprocessing is a string of pointers to spelling table entries, punctuated by those vocabulary words which open and close blocks in the block-structured symbol table, activating and deactivating meanings for those spellings.

RWORD: The RWORD Package performs the initial lexical step of breaking a continuous stream of characters in the input string into appropriate items, which constitute the syllables, words, and punctuation marks of a language. The package consists of two main parts: one constructs a finite-state machine to recognize a specified set of items, the other executes the machine at run time. In the current RWORD, each type of item is defined by a statement in a regular expression language. The statement describes the composition of that type of item in terms of individual

characters, or classes of characters. Output from the RWORD system-building package is a decision machine, which may be used to dispatch control to any number of arbitrary processes with great efficiency, using the finite-state-machine logic. At any given time the machine is in a unique state, during which it reads a single unique character from the input string, and then dispatches directly to a new state, performing some specified action during the transition.

AEDJR: AEDJR performs both syntactic and semantic parsing of an input statement, and in the process builds a First-Pass Structure, which is a model of the input statement. In this structure the context of each word is shown by its location in a binary tree, and also represented is the sequence in which the words and phrases should be considered to extract meaning. An appropriate Second-Pass operator can follow this indicated sequence to complete the translation by compiling a program, or building a model for graphical display, or doing whatever is required to solve the particular problem.

A great many modifications are being made to the existing AEDJR as it is prepared to be the front end of AED-1. A comment feature has been added so that input files describing languages may contain their own documentation. A quotation feature permits spellings of AEDJR command vocabulary words to be used in the languages being defined. A generalized attribute feature permits new features to be added to the AEDJR command vocabulary without requiring extensive rewrites of the system itself. The user may also tag certain words of his language with "dun bits" to indicate that once a proper parsing of these words has occurred, it may be assumed that the structure is correct. Any errors which may occur are restricted to lie in the area between two successive dun bits, and the parsing process may proceed outside that area.

C. SECOND PASS

During the past year, we resumed an intensive study of the entire second-pass problem and developed a new two-part scheme with some exciting possibilities: the first part figures out useful plans and how to select them, and the second actually performs the selection process at run time. When the complete system is operational, a straightforward transcription of the descriptions of machine registers and instructions from the manuals for a particular computer will result in the automatic generation of efficient second-pass programs to select plans to generate efficient machine code.

Index Register Assignment: One of the most intricate and potentially important areas of "selector" design for AED-1 concerns the algorithm for allocation of specific hardware index registers (or similar multiple

machine registers) to specific program variables. An extensive study of the sparse literature in this field was completed during the reporting period. This study and new research has led to a scheme which gives results very near to the optimum, and which also interlocks neatly with the environment vector and hint mechanism of the AED-1 second-pass scheme.

Delayed Merge: The delayed merge portion of the AED-1 second pass has been excised and modified considerably to yield a generalized package which will be useful in many parts of system building. The delay permits a set of arbitrary actions to be decided upon in one sequence (the generation sequence), but postponed for execution in a different sequence (the playback or execution sequence). The package includes automatic recursive buffer generation, so there is no limit on the size or complexity of the structures involved.

I/O Buffer Package: Although, of necessity, every existing programming system has programs to perform input and output of files of characters, there is no satisfactory generalized treatment of this intricate subject. We have attempted to derive an idealized total modeling plex (data, structure, and algorithm) that we can adapt to any given set of circumstances by specifying a suitable mechanization of the ideal. This study has resulted in a powerful and efficient partial package which will be suitable for the initial AED-1 effort, and fits all known existing file-structuring and operating-system features.

Features Feature: The Features Feature permits more precise selection of individual programs of a package than is allowed by the library feature of standard loaders. The writer of an AED-0 integrated package labels portions of the package source file with mnemonic names of "features" which each portion is designed to implement. The user then selects package procedures he intends to use by feature name and calls for a "features expansion" of the package source file, creating a new source file which contains only those sections tagged with the specified names. Compilation of that source file then results in an object file with only the desired procedures ready for loading.

Alarm Reporting: The primary design criterion for the new generalized alarm package is that a working program of the system should not be directly burdened with elaborate alarm reporting features. It should contain only the necessary tests and decisions to determine that an alarm condition exists, and the programming necessary to collect pertinent transient information from the current environment. This information is recorded at the time the condition is detected and is saved for later reporting of the alarm condition in full elaboration. Thus valuable core memory is not occupied by excessive amounts of verbal descriptions and formatting information required for nice error descriptions. The alarm

information can be reported in many different forms — one for the user, another for the system programmer, another for collecting the statistics on system behavior, etc. Also, the alarm system features can be used as a generalized system programming technique for providing elaborate control and intercommunication between various program phases.

D. CADET

CADET is intended to process both verbal and graphical language in the same system. It was demonstrated in 1964 that verbal and graphical language forms are subsets of the technique of the Algorithmic Theory of Language (in fact this relationship was an integral part of the initial development of the language theory itself). Attention now is on the programming of CADET-1, making maximum use of the facilities already present in AED-1.

In CADET, the First-Pass Algorithm operation, and the assembly of characters into words by the RWORD Package, are the same for both verbal and graphical processing. There are two sources of characters in the input string — verbal characters from the typewriter, and graphic characters (display coordinates, light-pen hits, etc.) from the display console. Mixed phrases containing both verbal and graphical inputs will be processed by the basic macro facility, augmented to provide type constraints on macro arguments and other generalizations.

Many observers have commented on the awkwardness of switching back and forth between light pen and typewriter actions in many of the applications of computer graphics now in existence. With type-constrained arguments in calling structures, it is possible to overcome this awkwardness in a very natural way. Although a great deal of work remains to be done in this study, it is apparent that our objective for CADET of an overall systematic model of the man-machine problem-solving process is both realistic and achievable.

E. DISPLAY INTERFACE SYSTEM

The general approach to the display interface problem is intended to yield a machine-independent, display-independent, problem-independent, operating-system-independent approach to coupling graphic displays to man-machine, problem-solving systems in time-sharing. During the reporting period the basic system for operating the ESL Display Console through the PDP-7 computer (attached to the data channel of the time-shared 7094) was made operational. The system structure uses a minimal executive residing in the PDP-7, augmented by additional PDP-7 programs to suit the user's needs. As a first step in meeting these needs, the PDP-7 is being programmed to provide an exact duplicate of the features of the present A-core module for the ESL Console, so that existing

display programs will run in buffered mode without change. These programs are all written and in process of final debugging (see also discussion under Display Systems Research).

F. AEDNET

During the reporting period the AEDNET* system of digital computer programs became fully operational. AEDNET simulates a wide class of non-linear electronic networks whose elements are non-linear, time-varying capacitors, resistors, inductors, and dependent and independent sources. It permits constant or variable step integration of the network differential equations.

A significant feature is the <u>device</u>, which permits the user to define a subnetwork, give it a special symbol, and use the result as a new network element in the construction of larger networks. A new version of AEDNET, developed for classroom and research use, replaces the graphical input-output with typewriter input-output to enable several students to use the system simultaneously.

G. AED COOPERATIVE PROGRAM

The AED Cooperative Program encompasses those aspects of the overall M.I.T. Computer-Aided Design Project effort which are sufficiently developed to merit industry participation. A major feature has been industry sponsorship of visiting staff members who work with the M.I.T. staff learning the capabilities of AED while contributing to its improvement. During most of the current reporting period, twelve visitors were in residence at M.I.T.

The First AED Technical Meeting was held at M.I.T., June 22-23, 1966, with 54 people from 24 organizations (plus 51 people from various M.I.T. groups) in attendance. Topics included the AED-1 Processor, the AEDJR System, and computer graphics.

The Second AED Technical Meeting was held at M.I.T., January 25-27, 1967, with almost 350 people from 90 organizations (plus 58 people from various M.I.T. groups) in attendance. The main emphasis of the meeting was a workshop on the use of the AEDJR system and AED-0 language for making special user-oriented systems.

A number of companies who have not participated directly in the AED Cooperative Program have also obtained copies of system releases and documents to experiment with AED-0 and AEDJR on their own problems. Copies of system releases of the AED project are available for use on the IBM 709, 7090, and 7094 computers. The AED system also is

---

* Partly supported by NASA under Grant NSG-496.

widely used in both the M.I.T. Computation Center and Project MAC time-sharing facilities.

On-Line Simulation of Networks and Systems - Michael L. Dertouzos

The main objective of this research is the effective use of present and projected on-line utilities in the design of electrical networks and systems. This includes research in the mathematical foundations of network and system analysis, the "linguistic" basis of network and system representation and the interactive features essential for the design of such systems.

The research activity of this group is divided into two main categories: networks, where relationships are implicit; and so-called block-diagram systems, where relationsips are primarily explicit. In the case of networks, the main emphasis is placed in the development of an integrated on-line circuit design system, CIRCAL-II. This evolutionary system and its predecessor, CIRCAL-I, are used as an experimental forum to test and incorporate research results on new computer-oriented network-simulation techniques. In the case of systems, an "equivalent" on-line simulator is under development in which analog or digital systems are simulated as compositions of primitive functions and functionals. In addition, a special class of digital memoryless systems, consisting of threshold elements, are being studied from both a theoretical point of view and in the context of on-line design.

A. CIRCAL-II DEVELOPMENT

On the basis of experience gained with the existing program (CIRCAL-I) for on-line design of electronic circuits, specifications have been established for the next generation of programs under the name CIRCAL-II. During the reporting period, the semantic structure and syntactic details of CIRCAL-II have been identified, commands for this system have been flow-charted, and a start has been made on implementing the program. This work is supported in part by NASA under Grant NSG-496.

To aid in achieving generality of the CIRCAL-II system, a number of relevant off-line and on-line circuit analysis programs were studied for capabilities, common features, and areas of strength and weakness. Off-line programs studied to date are: NET, ECAP, CIRCUS, and SCEPTRE. The on-line programs are AEDNET, CIRCAL-I and OLCA. In particular, a detailed study has been conducted of the temporal and memory-space requirements of AEDNET and CIRCAL-I. In addition, the use of variable time increments was studied in some detail.

Prominent characteristics being implemented in the CIRCAL-II system are 1) a "standard" input/edit package, 2) a "common" data structure with means for performing changes on it, 3) one or more network-analysis operators that can operate on this data structure, 4) a common output data structure, 5) a set of definitional commands, and 6) housekeeping and instructional commands.

Through the input/edit package, the user will be able to create or edit files, which may be circuits, defined elements (or "nests") as "macros" consisting of other elements, functions, functionals, or new system commands. Circuits, nests, and functions may also be inputed graphically creating corresponding display files. The purpose of this subprogram is to create files without particular regard to their ultimate use. Editing on a typewriter console is similar to the ED program of CTSS; on a graphical terminal, by a light pen and a typewriter in combination.

Files representing circuits and nests will then be processed by the data-structure-creating operator, which organizes elementary blocks of information in memory in one-to-one correspondence with the circuit. The purpose of forming this data structure is to establish a convenient representation from which a large class of analysis operators can "take over" and to which changes by the user or by the program can be easily made. Some features of this data structure are node-oriented and element-oriented access to network topology, and easy access to homogeneous network subclasses, such as all resistors, nests, linear elements, voltage-controlled capacitors, and so forth. In addition, the data-structure-creation process introduces diagnostic examination of the input file being processed for errors common to the analysis operators.

The analysis operators, each equipped with further diagnostics, will operate upon the data structure, generating results in the form of current and voltage waveforms, and other functions of current, voltage, time, node, and element, which are defined by the user. Usually, only one operation is invoked for a given analysis unless the network is "torn" into subnetworks such as "linear" and "non-linear". In that case, the analysis operator for the overall network does the tearing and will call, for example, the linear and non-linear analysis operators and then combine their results.

Results of the analysis operators are organized in a common output data structure from which standard or defined (by the user) plotting routines can display the results either graphically or on the typewriter. Alternatively, user-defined commands, by the DEFCOM package, may operate on these results and induce changes in the network and subsequent analyses, until some user-defined criterion is satisfied — this is the optimization capability of the program. The definitional commands of

CIRCAL-II are also capable of defining such entities as networks, nests, functions, and functionals (e.g., to represent hysteresis).

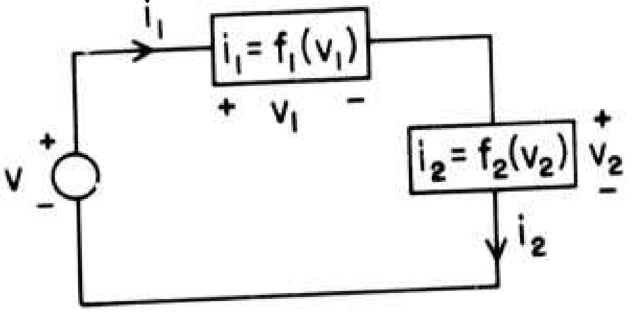
Finally, the housekeeping and instructional commands will provide information to the user of the type "Where am I?", "What is a function of a?", "What is command b?", as well as brief or detailed instruction-manual excerpts for operation of the system.

B. FUNCTIONAL INVERSION

The main objective of this work is to develop methods for rapid solution of systems of non-linear equations, typically arising in on-line network analysis. These solution methods differ from conventional iterative techniques in that some initial effort is expended to "invert" the system of equations (similar to matrix inversion) and thereafter, for any given parameters, the solution is obtained directly through interpretive evaluation (similar to multiplication by the inverse matrix) at relatively small computational cost. This work is supported by the National Aeronautics and Space Administration, in part under Contract NGR-22-009-158 and in part under NSG-496.

Toward this objective, an algebra of functions was developed which makes possible the representation of network response functions in terms of the basic non-linear network-element functions. The philosophy behind this functional representation is two-fold: first, such a representation permits functional inversion through an initial overhead in computing time spent setting up the desired response functions; second, a functional representation can be useful in the associated area of non-linear network synthesis.

The algebra under consideration has as its elements a class of monotonic and identity functions of one or more variables. The permitted operations use functional addition, subtraction, and extentions of the ordinary composition and inversion operations. As an example of functional inversion in the non-extended ordinary algebra, consider the network shown below.



Suppose we wish to determine (for interpretive evaluation) the function Φ such that $v_2 = \Phi(v)$. Writing KVL around the loop gives $v = \left\{I + f_1^{-1} * f_2\right\}(v_2)$. In this expression, I denotes the identity function $I(x) = x$, $(\ )^{-1}$ denotes ordinary functional inversion, and $*$ denotes ordinary functional composition. Thus the desired function Φ can be expressed as $\Phi = (I + f_1^{-1} * f_2)^{-1}$.

The above operations of ordinary composition and inversion are adequate to express the response function of any series-parallel network. For more general networks, the definitions of these two operations must be extended to functions of several variables. To date, it has been shown that the response of any network, or more generally, the solution of any set of simultaneous non-linear equations, can be represented in this extended algebra.

To make practical use of this algebra in on-line network analysis, it is necessary to develop methods of approximating functions in forms compatible with the extended and ordinary algebras, so that the operations of addition, composition, and inversion can be carried out efficiently. This topic is presently being investigated, and several canonic forms consisting of composition sequences of certain primitive functions have shown promising results.

C. TEARING TECHNIQUES

Separation or "tearing" techniques permit analysis of networks "a piece at a time", and also combination of these individual solutions into the solution of the overall system. Such techniques are especially useful for computer analysis of networks, since subnetworks with identical topological structure or nested components need only be analyzed once. In addition, networks can be torn into sections which are homogeneous in character (e.g., completely linear, completely non-linear, memory, or memoryless) and computational techniques most efficient for the particular network class can then be utilized. A third important (but not so obvious) reason for tearing is that solution of the total system through torn subsystems is generally much faster than direct solution of the overall system. This is especially true in the case of spare networks (where few elements are connected to each node). Finally, tearing can be used to separate components whose parameters are to be varied from those that will remain fixed from one analysis to the next. This results in computational savings, since only part of the system need be analyzed for each new setting of the variable-component parameters.

To explore these possible tearing approaches, a considerable amount of time was spent studying the pioneering work of G. Kron.* This led to development of a method for matrix partitioning, which in several instances is more efficient than the Kron tearing method. Tearing techniques were then applied in the time domain. In the case of linear systems analyzed through the state-space approach, although it was possible to express the state equations of the connected system in terms of the individual torn subsystems, it was not possible to use the subsystem solutions to obtain closed-form solutions for the overall system. The same problem was unsuccessfully attacked by transform techniques. Finally, by rephrasing the problem in operator terms, and by using a set of suitable approximating functions, closed-form solutions were obtained. In particular, linear networks (with memory) and non-linear one-element type networks with known terminal characteristics can be interconnected and analyzed as an overall system using the above techniques. Work in this area is in progress with examination of other tearing possibilities.

D. COMPUTER-AIDED SYSTEM ANALYSIS AND SIMULATION

In work being carried out for NASA under Contract NGR-22-009-158, the object is to obtain a digital computer program for <u>on-line</u> use in analysis and simulation of block-diagram systems. In many respects this program will resemble the present CIRCAL program for on-line design of electronic circuits. Systems to be simulated by this program are quite general: recursively defined block-diagram system representations, which are either "analog" or "digital", linear or non-linear, memory or memoryless. System components are characterized by a definite input/output relationship (i.e., a cause-effect relation). This makes possible the explicit description of non-linear functions, in contrast to networks where characterizing functions are implicit. In addition, the proposed program is able to handle systems containing implicit relationships or "loops". Techniques have already been developed for simulating such systems, usually by iteration.

A literature search has revealed that there exist at the present time about two dozen programs which simulate some subset of the general block-diagram system, and that these fall into roughly three categories: analog-computer simulators, differential-equation solvers, and discrete-system simulators. The majority of these programs have apparently

---

*Kron, G., "A Set of Principles to Interconnect the Solutions of Physical Systems," <u>Journal of Applied Physics</u>, Vol. 24, no. 8, August, 1953

Kron, G., "A Method for Solving Large Physical Systems in Easy Stages," <u>Proceedings of the IRE</u>, April 1954

arisen from the desire to simulate analog-computer operations on a digital computer. Almost all were developed for batch use, with occasional attempts to introduce some rudimentary on-line capabilities.

The proposed program will have provisions for defining elements which are either memoryless (functions) or memory (functionals), the domain and range of which may be either the reals or the set of two values, zero and one. Such systems will be amenable to editing, that is, changing the values of certain parameters or initial conditions, or changing the topological structure of the system itself. The method of analysis will be time-domain simulation through the state-space approach. Execution of the primitive functions and functionals will be through precompiled code, while higher-level system components will be executed interpretively.

Efficient analysis routines for such systems are highly dependent on the choice of an appropriate data structure. The program under consideration sets up a model in one-to-one correspondence with the actual physical system being simulated; in other words, a list-type structure or modeling plex is established. A measure of efficiency (time-storage product) was studied in the analysis of systems modeled by this structure. Two methods were proposed: system evaluation from inputs to output (forward analysis), and the opposite procedure (backward analysis). It was found that subject to a preliminary sort, the forward approach was considerably more efficient.

E. THRESHOLD-LOGIC NETWORK SYNTHESIS

The object of this work under NASA Contract NGR-22-009-158 is to generate computer-oriented techniques for threshold-element-network realizations of Boolean functions. Part of this threshold-logic network-synthesis problem is the non-unique decomposition of a non-threshold Boolean function F into another function G, so that the transformation from G and the independent inputs to F can be accomplished by a single threshold element. It has been shown that this decomposition is governed by a necessary and sufficient condition, expressed in terms of a finite set of extremal multi-dimensional vectors in a convex cone associated with function F. These extremal vectors have been investigated, some of their properties determined, two types enumerated, and some relationships among them found. Two computer-oriented approaches to decomposition have been introduced. One is a general approach which is based on a full knowledge of the set of extremal vectors and gives a full characterization of the family of decompositions. The other is a more practical approach which uses only partial knowledge of the set of extremals, and is consequently only a partial solution to the decomposition problem.

Another part of the threshold-logic network-synthesis problem is the decomposition of the desired Boolean function and decomposed subfunctions into typically two or three (for technological purposes) Boolean

subfunctions. This decomposition, in contrast to the foregoing approaches, is not conducted through a threshold element, but rather through use of any two- or three-input Boolean function, which is in turn realized by a tabulated threshold-element-network. Master's theses by M. Edelberg and Y. D. Willems were completed in connection with this work. (See Edelberg and Willems, Appendix B.)

Simulation Studies of Strapped-Down Navigation Systems on the PDP-6 - Frank B. Hills

In a research program for the Air Force under Contract AF-33(657)-11311, we have been studying the digital computation problems peculiar to "strapped-down" navigation systems. In such systems, inertial sensors (gyros and accelerometers) are rigidly mounted to an airframe, rather than being mounted on a stabilized platform, and the accelerometer outputs must therefore be transformed from body axes to the coordinates in which the navigation computations are made. We are using the direction cosine transformation. Because of the rotations of the vehicle, the direction cosines are not fixed, and hence must also be computed. This is done by numerically solving nine simultaneous differential equations in which gyro data are the independent variables.

Analytical studies made by the group have shown that major errors, insofar as the computations are concerned, are due to sampling and quantizing the gyro and accelerometer data. Because acceleration and rotation are non-commutative, and the components of rotation are also non-commutative, the effects of sampling and quantizing on the error are very complex. Therefore, simulation studies were started to supplement the mathematical investigation.

An outline of the desired simulation program has been given in previous Project MAC progress reports. Briefly, Project MAC's PDP-6 was chosen for the simulation studies because of its ability to perform double-precision arithmetic at high speed. (The reference calculation must check various algorithms against a required 46-bit accuracy.) Also, a major goal in writing the simulation program was organization which would permit a great variety of simulation tests with a minimum of debugging once the main program was written. The flexibility of the macro definitions allowed by the PDP-6 assembly program was of great assistance in this regard. During the past year, the availability of the 256K memory on the PDP-6 made possible a number of changes in the simulation program which

1. increased the amount of data that could be obtained from a simulation run,

2. increased the number of algorithms for attitude and position that could be tested, and

3. increased the number of functions that could be printed out to improve and extend the interpolation of navigation performance.

With the improved program, eleven complete simulation tests were run, each concerned with a different set of rotation rates and accelerations. Data was produced from each test to show the effect upon navigation accuracy of: updating the rate, quantization level of both rotation and velocity, algorithms, and computation round off.

The investigation has been successfully completed and a final report is being written.

BLANK PAGE

HARVARD UNIVERSITY

Generalized Desk Calculator

Interactive Social Psychology Experiments

Faculty

A. G. Oettinger   A. Ruyle   P. J. Stone

Staff

K. Winiecki

Generalized Desk Calculator - Anthony G. Oettinger and Adrian Ruyle

The use of CTSS by Project TACT (Technological Aids to Creative Thought) during the past year has been principally to maintain TOCS (TACT On-line Computing System), our technological aid generalized from the computing system of Culler. This work has been partially supported by ARPA Contract SD265.

Extensive use of this system for teaching, as projected in the last progress report, has been precluded by our slender time allotments. Accordingly, the storage-tube CRT scope terminal at Harvard University has been dismantled, and use of the TOCS on our problem number is restricted to demonstrations and short problems which do not strain our time resources.

To make TOCS available to other MAC users, it has been made read-only linkable by all problem/programmer numbers. The following commands by any MAC user will establish a working connection to the system:

```
LINK TOCS BCD T262 3491
RUNCOM TOCS
```

and thereafter the command:

```
R TOCS STDOPS
```

is sufficient to put him in touch with the system. The TOCS manual, TACT Memorandum 43, is available through the Project MAC Document Room.

Interactive Social Psychology Experiments - Philip J. Stone

(See "Man-Machine Communication", this volume.)

BLANK PAGE

MAN-MACHINE COMMUNICATION

ELIZA

Man-Machine Natural-Language Communication

Status Mobility and Syntax Analysis

An Interactive Inquirer

Academic Staff

D. A. Huffman J. Weizenbaum

Guests

S. Lorch	-	Massachusetts General Hospital
M. T. McGuire	-	Massachusetts General Hospital
G. A. Miller	-	Harvard University
G. C. Quarton	-	Massachusetts General Hospital
P. J. Stone	-	Harvard University

ELIZA - Joseph Weizenbaum

During the current reporting period a major deficiency in the ELIZA program discussed in last year's report was removed. It will be recalled that the ELIZA system is one which permits natural-language communication between a man and the MAC Compatible Time-Sharing System via a typewriter console and that the machine responses to the man's input are governed by a set of transformation rules called a script.

The difficulty with the early ELIZA system alluded to above was that the system could do no computation in any significant sense. It was, for example, impossible for that system to respond to the question "What is x squared?", assuming a value had previously been assigned to x with a correct number. In order to enhance the ability of ELIZA in this direction it was necessary to design and build an evaluator, that is, an interpretive program, to which computational and logical tasks could be given for execution. Such an evaluator was built.

The composition of this evaluator proved to be an interesting task with ramifications extending beyond its immediate utility as a subsystem to ELIZA. An effort was made to design the program in a way such that it would prove a useful tool for the teaching of a number of deep issues in the field of programming languages. The resulting program is essentially an interpreter version of SLIP with approximately the power of LISP but with very considerably simpler syntactic conventions than LISP. It pays most particular attention to the distinction between bound and free variables in various contexts and to the problem of retrieval of abandoned space (garbage collection). An unusual property of the system, one that it shares with LISP, is that functions defined within the system may deliver functions as their arguments. The fact that all functions within the system may be recursive leads to a very general form of the garbage collection problem. That problem is completely solved within the system. Furthermore, garbage collection proceeds in an incremental, or distributed, fashion as opposed to being an occasional cataclysmic event.

The new ELIZA system is the subject of a paper to appear in a forthcoming issue of the Communications of the ACM. It is currently in use at the Education Research Center of M.I.T. in a teaching-machine context.

Man-Machine Natural-Language Communication - Gardner C. Quarton, Michael T. McGuire, and Stephen Lorch

Two groups of 24 subjects each were tested in the following way. Subjects in the first group were told they were "conversing with a computer". Subjects in the second group were told they were "conversing with a person". Both groups conversed in unrestricted natural language

input to a "programmed investigator". Replies to subjects' input were similar to those outlined in Section C of our writeup in Progress Report III. A Latin Square design determined reply types, to check order and sequencing effects on subject ratings of replies for "evidence that the reply to their input gave evidence of understanding". Analysis of the results show that there was no difference in the response of the two groups. These results are interpreted to mean that even though subjects received different instructions, these had no effect on their basic "habitual" information-processing systems. It seems probable that when natural language is used in communicating with a computer the user will tend to use the same information-processing rules and make the same assumptions as he would when talking to a human. It may be necessary to program computers accordingly, since humans are inaccurate processors of information and appear to lack efficient disconfirming systems for their views.

A second and quite different study dealt with both the sequencing of information presented to subjects and the effect of the sequencing on subsequent discrimination in man-machine interaction. The results of this study show that when initial computer information is primarily interrogative in nature, subjects do not subsequently discriminate other types of presented information. When non-interrogative information is presented first, subsequent discrimination is possible. These studies are examples of an on-going attempt to establish experimentally the importance of those variables which apply to most potential users and which, in our estimation, may dictate specific programming features when computers are used in problem-solving by naive users.

Current studies may be broken down into three areas: those studying the relationship between display forms and subjects' recall (recall vs form) and usability in novel situations (usability vs form); studies of psychological theories of man-machine communication; studies which attempt to use the time-shared computer as a research tool in psychological studies. (Copies of published work or manuscripts in preparation are available through our offices at Massachusetts General Hospital upon request.)

Status Mobility and Syntax Analysis - Philip J. Stone

Present work is based on the formal conceptualization of father-to-son status mobility and on quantification problems of one aspect of English syntax. The problems share certain measurement and formalization properties which have made it useful to study them jointly. This has led to formulation of an orientation and techniques of measurement. The orientation is meant to make it increasingly possible for a social scientist to set out his theory and intuitions from which to design formal

concepts. This type work is made feasible by the liberal use of time-sharing computation in which the techniques (and the programs which implement them) are incessantly changed and evaluated while the data to which they are applied remains constant. Attention is given to development of programs which allow utilization of rapid man-machine interplay to rebuff and rebuild the analysis. Accessibility of the techniques in executable form has facilitated cooperation among researchers pursuing similar problems.

The time-shared facility has made possible a doctoral dissertation, Measurement in the Study of Intergenerational Status Mobility*, submitted to the Harvard University Department of Social Relations, by Joel Levine. The dissertation develops data transformations and measurement techniques that lead to a model of the mobility process. (The transformations, measures, and model are executed by programs that evolved with the dissertation.) It was found that some properties of the mobility data can be described by assuming a form of "constant interaction".

The syntax analysis similarly examines association properties, based on usage in text, in developing scaling techniques for syntactical properties. Briefly, this data included four prepositions and two articles in one text scaled in comparison to their behavior in a second text - one scale for each of the 15 pairs of words.

Next year's work will emphasize analytical and display techniques for more complicated forms of these problems.

An Interactive Inquirer - Philip J. Stone

We are concerned with a form of automated language processing called content analysis, and we are also interested in some new ventures in dialogues with the computer as applied to education. Content analysis procedures are concerned with the identification of repeated symbols or themes in text. These procedures have been shown to be relevant to research in psychology, sociology, political science, anthropology, and education.

An example of automated content-analysis scoring, in this case scoring for need-achievement, is seen in Figure 17. In this figure, the text appears on the left and the categories into which words and phrases are assigned appear on the right. A total evaluation is printed at the end of the story.

---

* Also supported by N.S.F. and I.B.M. fellowships and N.S.F. research funds.

Sentence 1: The student is dreaming about becoming a great inventor.	NEED TO-BE ADJECTIVE-POSITIVE ROLE POSITIVE SENTENCE SUM = A1
Sentence 2: After years of labor the crucial moment arrives.	TIME VERB-POSITIVE SENTENCE SUM = U1
Sentence 3: He hopes everything will turn out well.	NEED VERB-POSITIVE ADVERB-POSITIVE SENTENCE SUM = A1
Sentence 4: But the experiment will fail.	VALUE-POSITIVE FAILURE SENTENCE SUM = U1
Sentence 5: Displeased but still confident he will modify his procedures and try again.	AFFECT-NEGATIVE VALUE-POSITIVE SENTENCE SUM = A1

****SUMMARY****THIS DOCUMENT CONTAINS ACHIEVEMENT IMAGERY.

Figure 17. A Story Scored for Need-Achievement

The set of computer programs which performs this scoring is called the General Inquirer, and can be considered analogous to a very efficient clerk lacking any ideas of his own. Once it is told what to do, the task is carried out efficiently and mechanically. The direction for scoring co-occurrence patterns must be supplied in the form of rules. As a whole, we are quite pleased with our initial successes in scoring need-achievement. When 240 thematic apperception test (TAT) compositions were categorized by the computer (in batches of 60 stories), the percent of agreement between the automatic method and trained scorers varied from 82 to 86 percent. Since our purpose is to draw statistical conclusions, such as the "members of group X tend to have more need-achievement in their TAT's than the members of group Y", we can tolerate a certain amount of error in our measurement procedures.

The system has been successfully run at a number of different computer installations in the United States and Europe. From some 30 studies using the General Inquirer, we now have approximately six million words on punched cards, representing the kinds of language data that behavioral scientists tend to study. From these six million words, we have taken a sample of 500,000 words and put them in a massive "key-work-in-context". This listing informs us what word usages are most common, and often suggests contextual procedures for identifying them. Often, satisfactory rules can be identified in a few minutes. A very complicated and common word may take several days to work out. The task of examining several thousand words is long and tedious, but we hope to have a considerably improved accuracy in our mappings within a year or so.

Our initial experience with scoring as an interactive process began when we put our need-achievement scoring system on the MAC system. Figure 18 presents an example protocol from one subject. In this case, the subject is seated at the teletypewriter console and the directions for writing the story are presented to him by the computer. The subject then types his story. As soon as he is finished. he presses the return key on the typewriter twice and the computer immediately gives him an analysis of the story, first giving a summary of the amount of need-achievement present, and then giving a sentence-by-sentence analysis (for each of the four sentences in the story), showing where in the story need-achievement was found.

It is not difficult for a person interacting with the computer like this to quickly learn what kinds of stories will be scored by the computer as examples of need-achievement. For the complete novice, we might expand our directions and include some initial examples of what are and what are not achievement themes. As the subject types a number of

```
r nach
W 1529.8
```

Directions typed by computer

```
TAT STORY-SCORING PROCEDURE

WRITE A SHORT STORY TO THE PICTURE TO YOUR LEFT.
DESCRIBE WHAT IS HAPPENING, WHO THE PERSON IS,
WHAT HAPPENED IN THE PAST, WHAT IS BEING THOUGHT,
WHAT IS WANTED, WHAT WILL HAPPEN.  PRESS
RETURN KEY TWICE WHEN YOU HAVE FINISHED YOUR
STORY.
```

Story typed by subject

```
This man wants to become a doctor.  He works very hard to
get enough money to go to medical school.  He finally gets
in and fails all his courses.  Unhappy, he drowns his
sorrows in drunkenness.
```

Analysis typed by computer

```
**SUMMARY**  THIS DOCUMENT CONTAINS
ACHIEVEMENT IMAGERY.  SUBCATEGORIES ARE...
NEED = 1 INSTRUMENTAL ACTIVITY = 1 GOAL ATTAIN-
MENT = 1  GENERAL BLOCK  = 1 AFFECT = 2
THEME = 1

SUMMARY EACH SENTENCE.

1)   NEED + TO-BE + ROLE-POS + N-ACH-PRESENT +
2)   VERB-POS + ADVERB-POS + ADJ-WEAK-POS +
     INST. -AID + N-ACH-PRESENT
3)   FAILURE + NO-ACH +
4)   AFFECT-NEG + AFFECT-NEG + N-ACH-PRESENT +

END OF THIS ANALYSIS, READY FOR ANOTHER STORY
```

Figure 18. Need-Achievement Scoring: Interactive Procedure on a Time-Shared Computer Typewriter

stories, the computer could easily check whether they are only a stereotyped subset of the larger variety of possible need-achievement themes. If this were the case, the computer can then give some broader possible examples, and encourage the subject to try a wider variety of stories.

Our interactive scoring fulfills some of the basic elements of teaching-machine principles. The feedback is immediate and at the end of each story, a logical point for feedback. But note a fundamental difference. In most instances of teaching machines, the machine offers the subject a small, fixed number of alternatives and the subject must choose from one of these. Here, the role of dominance in interaction is much more balanced by having sets of alternatives several orders of magnitude larger than a typical teaching machine. The subject is typing a story of his own choosing, and the machine is responding to the subject. One does not have to watch many subjects to know that there is considerable pleasure for the student in having the machine respond to <u>his</u> ingenuity, rather than vice-versa. This reversal of initiative and dominance in the role of interaction emphasizes that: rather than "load" the student with information, the system should be designed to develop his abilities of reason and inquiry. One key factor in making this possible is that the subject has the freedom to express himself in his own native language at the typewriter.

As a further example of who leads whom in the dialogue with a computer, consider the interaction in Figure 19, where the computer is playing the role of a psychotherapist using the ELIZA system developed by J. Weizenbaum. Here the computer is responding to subject matter presented by the subject, offering direction when it fits into the context of the discussion.

Such conversational dialogues can require rather complex language analysis. A first step is the categorization of words and phrases, including the necessary contextual identification routines such as we are developing for our content-analysis dictionaries. Various pattern routines are then needed to identify the message, including a provision for the computer to make inquiries for clarification, should it be unable to make a satisfactory classification. Additional steps are needed to maintain a satisfactory logic of conversation. Often it is necessary that the computer create a file of attitudes expressed by the subject so that it can check for inconsistencies, or refer back to incomplete topics, or ask about relationships between topics. Often, if the dialogue ranges across topics, it is necessary that the computer's response set be adequately compartmentalized into topics, so the dialogue does not cross topics at inopportune times. Responses should often maintain the actual words employed by the subject; for example, the computer in Figure 19 uses the patient's actual words "boyfriend" and "depressed" rather than alternative words with the same meaning.

men are all alike
IN WHAT WAY
they're always bugging us about something or other
CAN YOU THINK OF A SPECIFIC EXAMPLE
well, my boyfriend made me come here
YOUR BOYFRIEND MADE YOU COME HERE
he noticed that i'm depressed much of the time
I AM SORRY TO HEAR YOU ARE DEPRESSED
he's right. i am unhappy
DO YOU THINK COMING HERE WILL HELP YOU NOT TO BE UNHAPPY
i need some help, that much is clear
WHAT WOULD IT MEAN TO YOU IF YOU GOT SOME HELP
perhaps i could learn to get along with my mother
TELL ME MORE ABOUT YOUR FAMILY
my mother takes care of me
WHO ELSE IN YOUR FAMILY TAKES CARE OF YOU
my father
YOUR FATHER
you are like my father in some ways
WHAT RESEMBLANCE DO YOU SEE
you are not very aggressive, but i don't think you want me to notice that
WHAT MAKES YOU THINK I AM NOT VERY AGGRESSIVE
you don't argue with me
WHY DO YOU THINK I DON'T ARGUE WITH YOU
you are afraid of me
DOES IT PLEASE YOU TO BELIEVE I AM AFRAID OF YOU
it pleases my father to think i am afraid of him
WHAT ELSE COMES TO MIND WHEN YOU THINK OF YOU FATHER
bullies
DOES THAT HAVE ANYTHING TO DO WITH THE FACT THAT YOUR
BOYFRIEND MADE YOU COME HERE

J. Weizenbaum
Harvard Review, Spring, 1965

small letters: young lady
CAPITAL LETTERS: COMPUTER

Figure 19. A Distraught Young Lady Consults the DOCTOR

In many instances, the computer typewriter can completely reverse its role from that of dominant teacher to that of instructive assistant. One such application is where the computer is able to respond to the context of a series of commands in natural language. A number of attempts have been made to have the computer respond to natural language commands. Similarly, attempts have been made to have the computer respond to the questions of a doctor diagnosing a disease or a Sherlock Holmes cross-examining a witness. Figure 20 presents a set of directions from a subject, using Weizenbaum's latest "ELIZA/OPL" system. In this interaction, the computer is contextually responding to the commands of the subject. Such kinds of learning become another resource in the wide variety of possible interaction patterns.

How good is a conversation with a computer? In certain cases it can be deceptively convincing. An oft mentioned criterion is known as "Turing's test". Turing's test has both a simple and a complex form, but let us consider a simplified situation. Consider a person sitting at a typewriter not knowing whether his typewriter is connected to a computer or to another person sitting at a typewriter. Can the person tell whether he is communicating with a human being or a computer? Actually, telling them apart can become difficult. McGuire, when he put subjects through a full hour of therapy with the computer, found that 62 percent of the subjects were convinced they were talking to a person at the other end, 21 percent were uncertain, and only 17 percent believed they were communicating with a machine. But we should share a little secret that deceived some of the most sceptical subjects. Usually when a typewriter is being controlled by a computer, it types evenly and rapidly like a mechanical teletype. Weizenbaum and McGuire arranged to have the machine type hesitantly and irregularly, to make occasional errors and back up to fix them, all at a speed of a very amateur typist. This one trick in itself was quite convincing that there must be a person at the other end. Occasionally the computer makes inappropriate remarks. Human subjects, however, seemed surprisingly willing to overlook occasional stupidities from the computer. It should be said that the quality of the computer's responses as a whole were usually quite satisfactory.

The development of interaction procedures using natural language will require the cumulative contributions of many people. Just as our development of "disambiguation" routines described above is expected to serve a number of different content-analysis dictionaries, so too, the

please tell me what the sum of 5 and 4 is

IT'S 9.0

good, call that a

VERY WELL

can you tell me what the product of a and 5 is

IT'S 45.0

call it b

VERY WELL

let f(x,y) be x**2+y**2

ALL RIGHT

please compute f(a,b)

IT'S 2106.0

what is the product of 45 and 45

IT'S 2025.0

what is the difference between that and a

IT'S 2016.0

call this z

J. Weizenbaum

Sept. 1966

small letters: student
CAPITAL LETTERS: COMPUTER

Figure 20. Student Giving Directions to the Computer Using ELIZA/OPL

increased sophistication of many-to-few categorization procedures can be drawn on by different investigators in developing more and more complex pattern analyses. The task is too complex for each investigator to start from scratch simply with raw text and a raw computer. Instead, he will need to cumulatively borrow on the previous work done by others. Such borrowing is essential if we are to get on with our work and to focus on issues rather than details.

Our interaction mode, in Figure 18, is based on the assumption that the subject can type, and at this stage of the game our typewriter happens to be attached to one of the most powerful and sought after computers in the country, Project MAC. Essentially, we are ready for demonstrations and limited research projects: not mass scoring of language material.

On the other hand, prospects for the future are rapidly improving. What took us an hour to process on the IBM 709 computer in 1961 can now, with improved programs, be processed in less than three minutes on the IBM 7094. As computer memories become larger, we will not only be able to monitor one or two complex scoring procedures at once, but also make a large variety of continuous monitorings. The script may call for a lesson in geography; but while the lesson is going on, continual scorings can be made by the computer of negativity, redundancy, and other cues from the student to decide whether the computer should break out of the role of geography teacher, and consider some other aspects of its relationship to the student. The student does not simply interact with a segmented series of programs, but rather develops an interactive relationship with the machine, with the machine building up an extensive file about the student from their past dialogue.

While the computer can analyze language information, while it can make inquiries when information comes in unanticipated ways, while it can construct maps of incoming information and make checks for inconsistencies, while it can adjust its vocabulary to the vocabulary of the person to whom it is talking, while it can compose syntactically complex answers, we certainly would not want to give the machine credit for being original or "understanding" in any sense of the word. Or would we? The machine as a teacher will always have its weaknesses. But perhaps we can expect children to be as tolerant of machines as they are of us adults.

BLANK PAGE

PROGRAMMING LINGUISTICS

Introduction of New Programming Linguistics

Curriculum Development In Programming Linguistics

The PAL-ISWIM System

The BCPL Compiler System

Student Use of a Time-Sharing System

Academic Staff

A. Evans, Jr. P. J. Landin J. M. Wozencraft

Non-Academic Research Staff

M. Richards

Research Assistants and other Students

T. J. Barkalow J. H. Morris, Jr.

Introduction of New Programming Linguistics - John M. Wozencraft

The Programming Linguistics Group, formed a year ago, has had two objectives: isolating various linguistic facilities underlying the specification of algorithms, and organizing this material into an introductory course for undergraduates who plan to major in computer science. These objectives are mutually reinforcing: without research, subject development would not be possible; without subject development, discovering the strengths and weaknesses of a proposed formulation would be far more difficult. Accordingly, the work has been supported jointly (and in nearly equal measure) by Project MAC and the Electrical Engineering Department.

The group's principal accomplishments to date consist of first-draft class notes and a programming language, called PAL, which has been implemented on CTSS at both Project MAC and the Computation Center. The notes and PAL were made available to a trial section of 20 sophomores and juniors during the Spring semester of 1967. We expect to refine the notes and increase the efficiency of the PAL implementation during the coming year.

In programming linguistics, the approach followed is intimately related to the branch of mathematical logic known as the λ -calculus. The relevance of λ -calculus to programming was first explored in depth by P. J. Landin, and most of our work thus far has been based on Landin's research. Remaining topics which deserve further investigation concern data structures and the fitting together of algebraic and imperative linguistic features in a more natural union.

Curriculum Development In Programming Linguistics — Arthur Evans, Jr.

The new sophomore course in the area of programming linguistics (see Wozencraft, this section) requires a student interactive system to process the language being taught. This language, called PAL, has been implemented as a CTSS program and has been embedded in a special interactive control system which serves three purposes:

1. An environment is provided so that students may test problem solutions using data sets supplied by the instructor. Further, the students' answers are checked by checking programs supplied by the instructor.

2. The system keeps records in grading files of the student's progress, so that the instructor may know how each student is doing.

3. The system provides security so that each student is insulated from the effects of the others and is further unable to access the solutions produced by the other students.

During the Fall semester, the prototype for this system was implemented on CTSS at Project MAC. The final version was then carried to the Computation Center where it was used by students. The processor portion and the student environment of the PAL language are the work of graduate students James Morris and Thomas Barkalow, respectively.

The PAL-ISWIM System – Peter J. Landin and James H. Morris, Jr.

Landin's programming language ISWIM and its immediate derivative PAL were implemented on CTSS, a translator-interpreter system.

ISWIM is modelled after Church's λ-calculus. The ISWIM expression

$$f(3) + f(4) \ \underline{\text{where}} \ f(x) = 2*x$$

corresponds to the λ-calculus formula

$$[\lambda f.\ f(3) + f(4)]\ [\lambda x.\ 2*x]$$

which, in turn, corresponds to the LISP S-expression

```
( (LAMBDA (F) (PLUS (F 3) (F 4)))
  (FUNCTION (LAMBDA (X) (TIMES 2 x ))) )
```

The interpretive mechanism is virtually the SECD machine.* The translator from ISWIM to λ-calculus is based on Floyd's scanner for precedence grammars extended to provide good error diagnostics. A primitive text-editing system is provided within the system to allow immediate correction of programs, although programs may also be entered via standard CTSS line-marked files.

The system was written in LISP to allow easy modification; a second implementation, written in a compiled language, is being planned. PAL was used as an instructional aid in an undergraduate course in Computer Science.

The BCPL Compiler System – Martin Richards

This compiler was originally developed to implement the language CPL (Combined Programming Language) on a number of different machines. The central feature of the system is the bootstrapping language BCPL.

BCPL is derived from CPL by removing those features of the full language which make compilation difficult: namely, the type and mode matching, and the variety of definition structures and their associated scope conventions. The large number of conditional commands and expression operators do not present a problem and are all included.

* P.J. Landin, A λ-Calculus Approach, Advances in Programming and Non-numerical Computation, Pergamon Press, 1966

The BCPL compiler was itself written in BCPL and serves as an example of the recommended style of programming for the other compilers of the system.

Great care has been taken to ensure that the BCPL compiler is as machine independent as possible. To demonstrate this, it is being implemented on the following five machines: the IBM 7094, 360/65, KDF9, Atlas 2, and GE 645.

Student Use of a Time-Sharing System – Thomas J. Barkalow

During the Spring of 1967, an operating system was developed to reduce the cost of allowing students to use CTSS. The programming exercises assigned in undergraduate computer subjects generally require the writing and debugging of short programs. This type of work may be done very conveniently on a time-sharing system. Thus an instructor with access to a time-sharing system, such as that at M.I.T., may be tempted to add each student in his class to the list of time-sharing system users by obtaining a programmer number for each student and a problem number for the subject. The instructor will soon discover, however, that the cost of this is prohibitive for most classes (approximately $1000 per month in overhead alone for a class of twenty at the current Computation Center prices).

A large portion of this cost is due to the overhead associated with adding a new programmer number to the time-sharing system for each student. It is possible to reduce this overhead by allowing several students to share the same programmer number, but this leads to three major difficulties:

1. A student may not be able to use the computer because another student is already using the programmer number to which he is assigned;
2. One student in a group may use an unfair portion of the group's time allotment;
3. One student in a group may read, modify, or delete the files belonging to other students in the same group.

These difficulties may be overcome, for the time-sharing system at M.I.T., by using certain of the system's facilities in conjunction with a special program. This program was developed and tested on the facilities at Project MAC, and used by a class of twenty students on the Computation Center's system. A similar system will probably be used for students taking Course 6.231 during the Spring of 1968.

BLANK PAGE

RESEARCH LABORATORY OF ELECTRONICS

Introduction

Cognitive Information Processing

Computer Analysis and Display of Beam-Plasma Instabilities

Computer Simulation of the Beam-Plasma Interaction

A Solution of a Nonlinear Differential Equation

Numerical Solutions to Wave Propagation Problems

The Cauchy-Integral Root-Finding Method

Modeling of Speech Production

The Psychophysical Reality of the Distinctive Features of Phonology

Computer Aids for Linguists

Current Distributions on Equiangular Spiral Planes

A Sinusoidally Stratified Model of the Insect Corneal Nipple Array

Blood Cell Classification

Touching and Overlapping Chromosomes

Private Libraries for TIP

Programming Support and Development

Academic Staff

G. D. Bernard
H. A. Haus
H. G. Matthews
A. Bers
W. L. Henke
K. N. Stevens
W. L. Black
D. Klatt
D. E. Troxel
R. J. Briggs
F. F. Lee
H. J. Zimmermann
S. C. Brown
S. J. Mason

Non-Academic Research Staff

E. F. Brown
G. Fratar
M. M. Pennell
H. S. Davis
J. Harwitt
E. C. River
C. L. Fontaine
E. C. Isaacs
O. J. Tretiak

Research Assistants and other Students

R. Bartsch
B. R. Kusse
C. L. Seitz
S. Brueck
D. T. Llewellyn-Jones
M. G. Smith
S. L. Chou
J. A. Mangano
C. E. Speck
J. A. Davis
R. R. Parker
I. T. Young
K. T. Gustafson
H. M. Schneider

Introduction

This interdepartmental laboratory provides facilities for academic research in three categories, designated as general physics, plasma dynamics, and communications sciences.

Work reported herein was supported principally by the Joint Services Electronics Program of the Army, Navy, and Air Force, with additional support from the Atomic Energy Commission, The National Science Foundation, and the National Institutes of Health.

The facilities at Project MAC were utilized extensively in a substantial number of R.L.E. research projects. These faculty and student research activities span a wide range of scientific and engineering subjects.

Cognitive Information Processing - Donald E. Troxel

Information processing of work performed by C. J. Tretiak (See Touching and Overlapping Chromosomes, this section.) required the largest amount of computer time. The results of this report cover two other problems which were solved using Project MAC facilities. Work was supported principally by the National Institute of Health, Grant 1-P01-GM-14940-01, and by the Joint Services Electronics Program, Contract DA-28-043-AMC-02536(E).

The first problem concerned the calculation of the optimum placement of two photomultiplier tubes in a flying spot opaque scanner. The tubes must be placed outside the bounds of specular reflection yet close enough to collect enough light to yield a sufficiently large signal-to-noise ratio and produce a tolerably flat field.

A program was written to solve the equation, for output signal strength as a function of input spot intensity, PMT position, rotation, and separation. The results were normalized to the center value of position and then tabulated.

Analysis of the resulting data showed that there is an optimum position for the PMT which would provide a flatness of field less than 3.5 percent and that minor variations from this would result in variations greater than 10 percent.

In the second problem a determination was made of the number of distinct secondary assignments for sequential machines with various numbers of states, assuming a minimum number of secondary variables. The primary purpose of the calculation was to demonstrate the futility of searching for minimum secondary assignments by denumeration even with a computer. For 2 through 11 states the number of distinct assignments are 1, 3, 3, 140, 420, 840, 840, 10810800, 75675600, and

454053600, respectively. Exhaustive searches are clearly impractical above 8 states, are questionable above 4 states, and can be done by hand with 4 states or less.

Computer Analysis and Display of Beam-Plasma Instabilities - Sun Lin Chou and Abraham Bers

This report covers the use of the Project MAC ESL Display to locate and follow saddle points of a dispersion equation in the complex plane. The dispersion equation describes the interaction of a thin electron beam with the ions in a warm electron plasma filling a cylindrical waveguide, and confined by an axial magnetic field. The dispersion equation is a polynomial of the form

$$\sum_{i=1}^{4} \sum_{j=1}^{6} A_{ij} k^{i} w^{j} = 0$$

where w is the radian frequency, k is the wavenumber, and the coefficient $\{A_{ij}\}$ are functions of beam-plasma parameters. Support for this project was partially supplied by the United States Atomic Energy Commission Contract AT (30-1)-1842.

According to the Bers-Briggs criterion[1] an absolute instability is indicated when two roots of k (obtained by solving the dispersion equation) merge from opposite halves of the complex k-plane as the imaginary part of w is increased to zero from some large negative value. This double root is observed as a "saddle point" when the complex w-plane is mapped onto the complex k-plane according to the dispersion equation.

Maxim G. Smith has written a program (named FOLLOW), applicable to polynomial dispersion equations up to order 12, which maps the complex w-plane onto the complex k-plane (and vice-versa), and also locates double roots in the latter plane (see Smith, Appendix B). Both complex planes are graphically displayed on the Project MAC KLUDGE scope. The program FOLLOW uses an iterative procedure which converges to the desired double root when given a sufficiently close initial guess.

Figure 21 shows a map of the real k-axis onto the complex w-plane for the dispersion equation under consideration. The existence of complex roots of w for values of Real(w) between 7 and 8 indicates that an instability is possible in that frequency range.

[1] Richard J. Briggs, Electron Stream Interaction with Plasmas, M.I.T. Press, Cambridge, Massachusetts (1964).

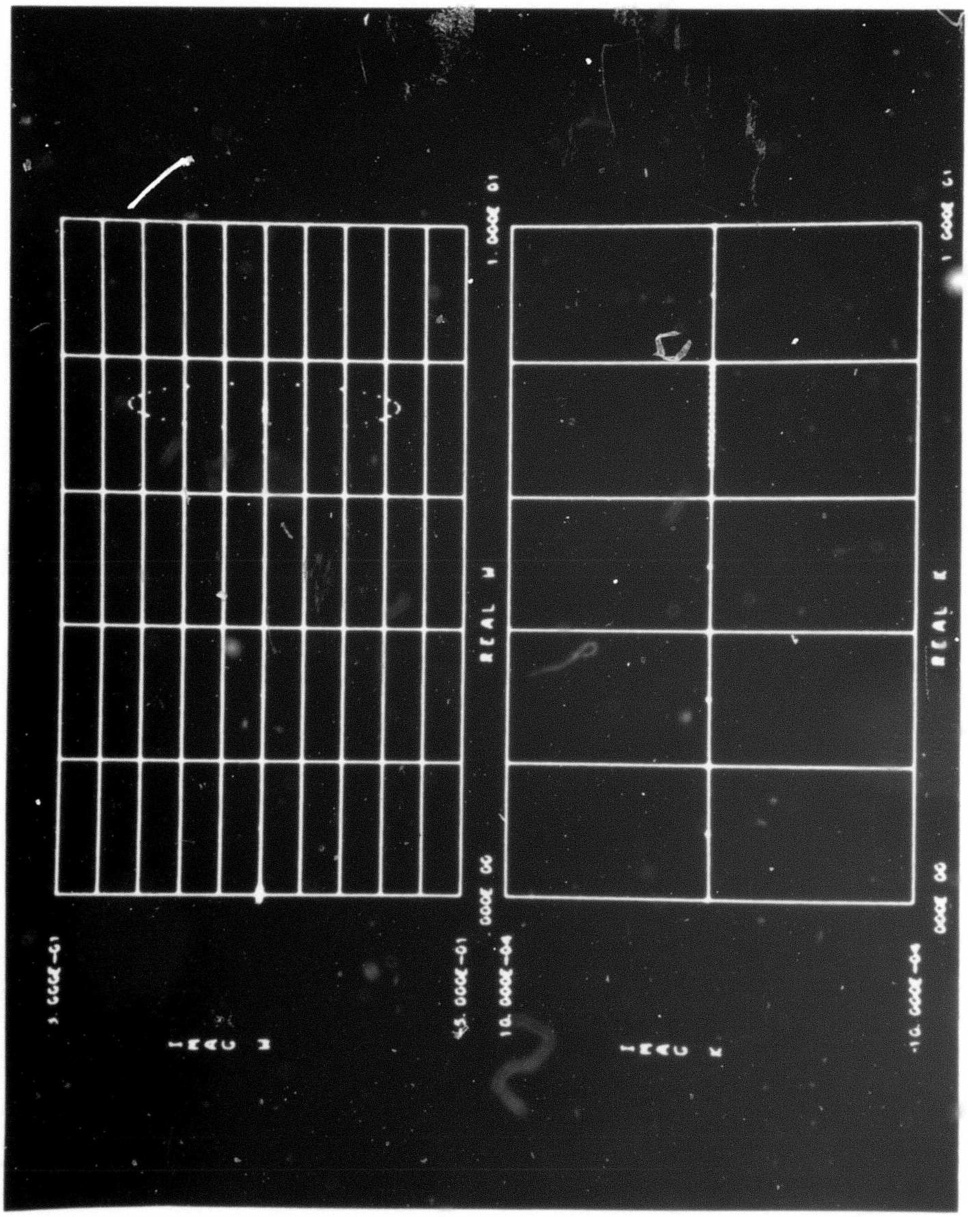


Figure 21. Map of the Real k-axis onto the Complex w-plane

In Figure 22 the loci of roots in the k-plane are plotted for two closely spaced values of Real(w) as Imag(w) is increased to zero from a large negative value. The characteristic "saddle" formation in the complex k-plane suggests a double root near the center of the saddle. It is noted that one pair of loci in the k-plane has crossed the real k-axis, thus satisfying the necessary condition for an absolute instability.

The approximate location of the center of the saddle was estimated visually and given to FOLLOW as an initial guess. The exact location of the saddle point was then computed by FOLLOW and displayed (Figure 23). The numerical value of k at the saddle point and the corresponding value of w were available as printed output.

Having found the correct saddle point for a certain set of parameters, FOLLOW was then able to trace the movements of the saddle point as a parameter was varied. For example, Figure 24 shows the motion of the saddle point in the k-plane as electron beam density is decreased. The growth rate of the instability (given by the negative imaginary part of the corresponding w for each saddle point) is seen to decrease and ultimately vanish as the beam density falls below a certain critical value. The absolute instability illustrated in the figures arises from the interaction of the slow (negative energy) synchronous beam wave with the plasma waves.

<u>Computer Simulation of the Beam-Plasma Interaction</u> - Jon A. Davis

Research is being conducted on steady-state non-linear beam-plasma interactions, with contractual support from the National Science Foundation, under Grant GK-1165. Our goal is to explain experimental measurements of collected beam velocity distribution by computer modeling. The beam-plasma system is represented by an electron beam disk model injected into an infinite plasma whose response remains linear. Interaction models with a homogeneous plasma have produced much wider velocity spreads than observed experiments. The introduction of longitudinal (i.e., along the beam flow direction) density gradients into the model has allowed considerably better agreement with experimental data. The experimentally determined collected beam velocity distribution is shown in Figure 25.

HOMOGENEOUS PLASMA AND ONE-DIMENSIONAL BEAM

We first attempted to model the beam with sheets, but the plasma was assumed cold and linear, and hence was treated analytically. A sheet with charge per unit area Q moving through the plasma creates a wake given by

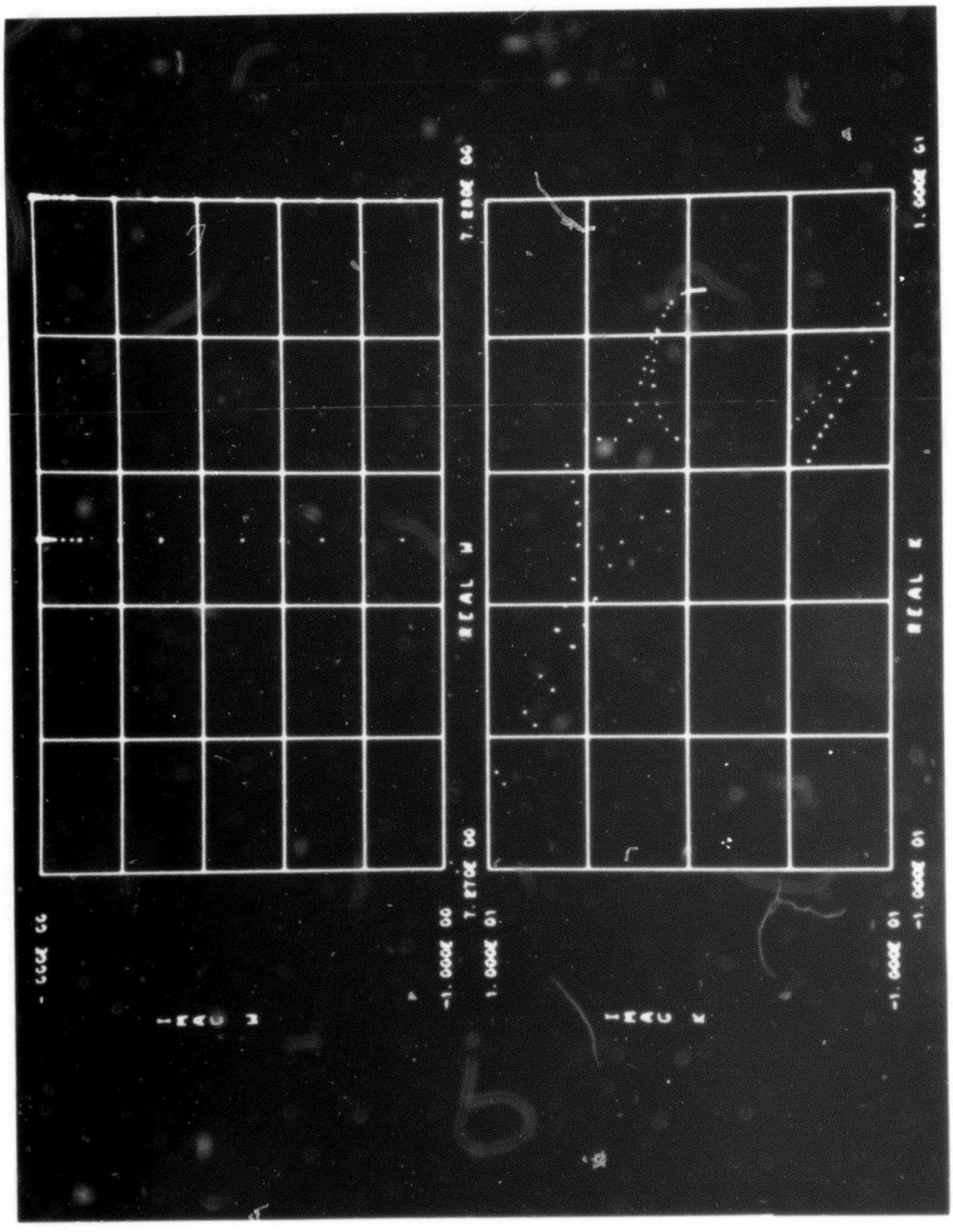


Figure 22. Plot of the Loci of Roots in the k-plane

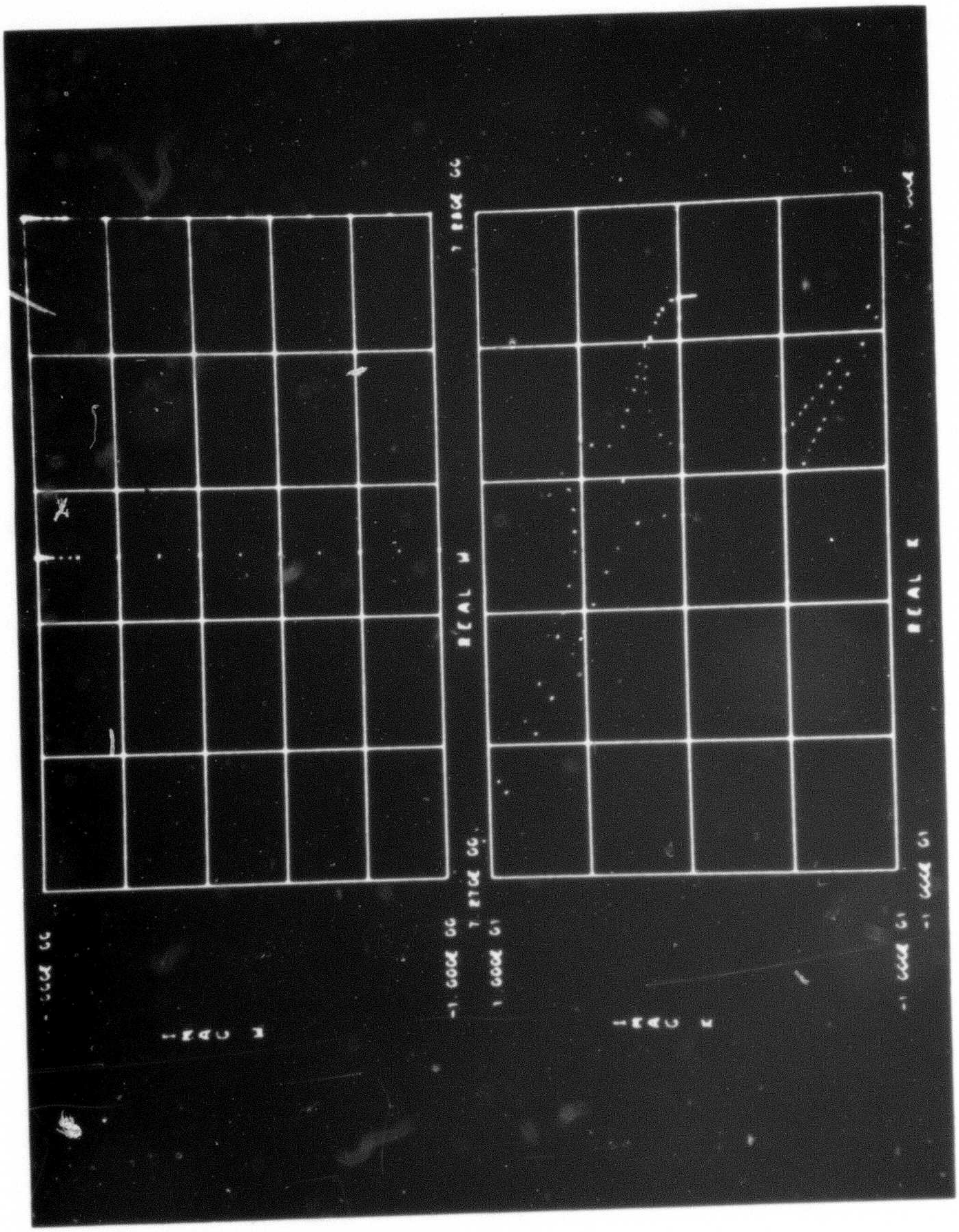


Figure 23. Saddle Point Location

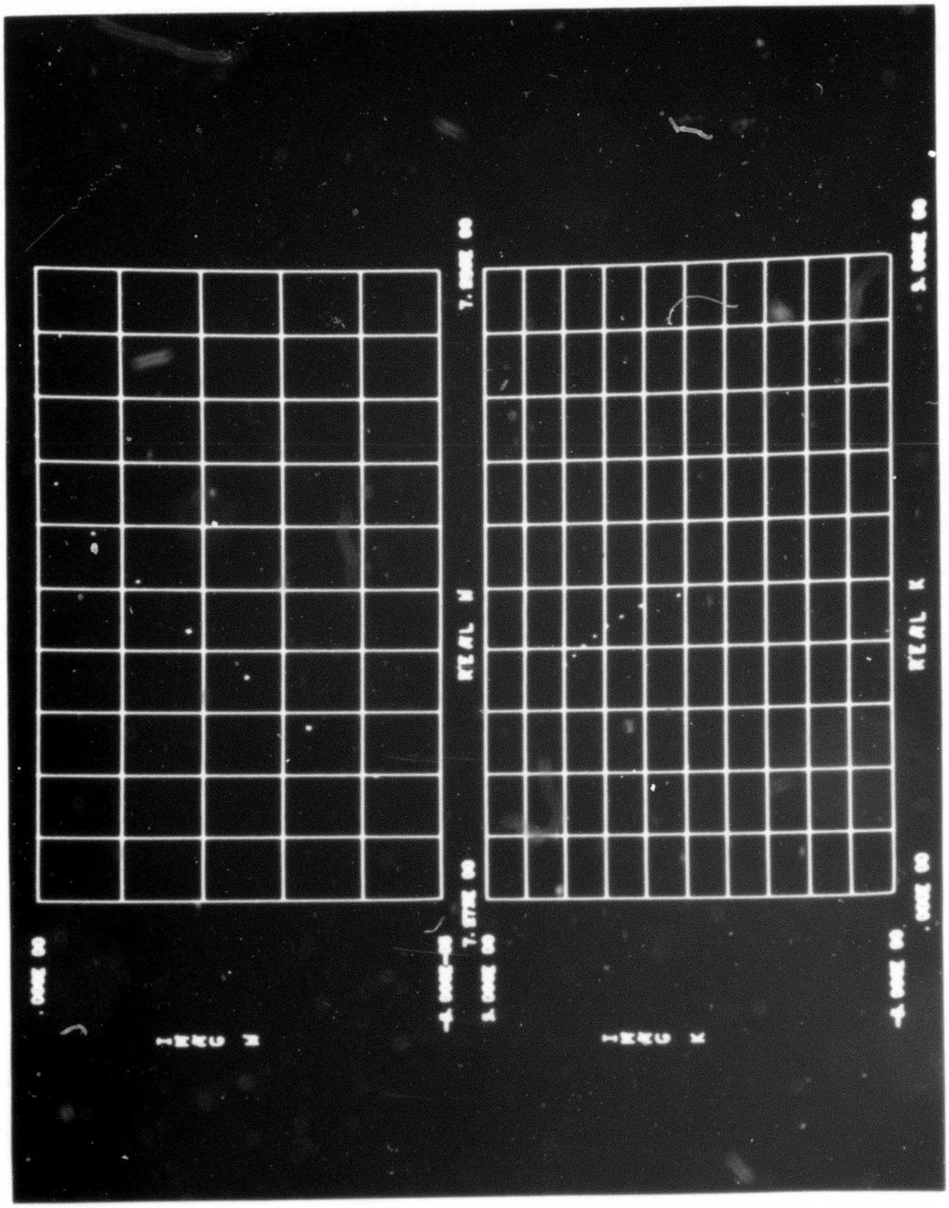


Figure 24. Motion of the Saddle Point in the k-plane

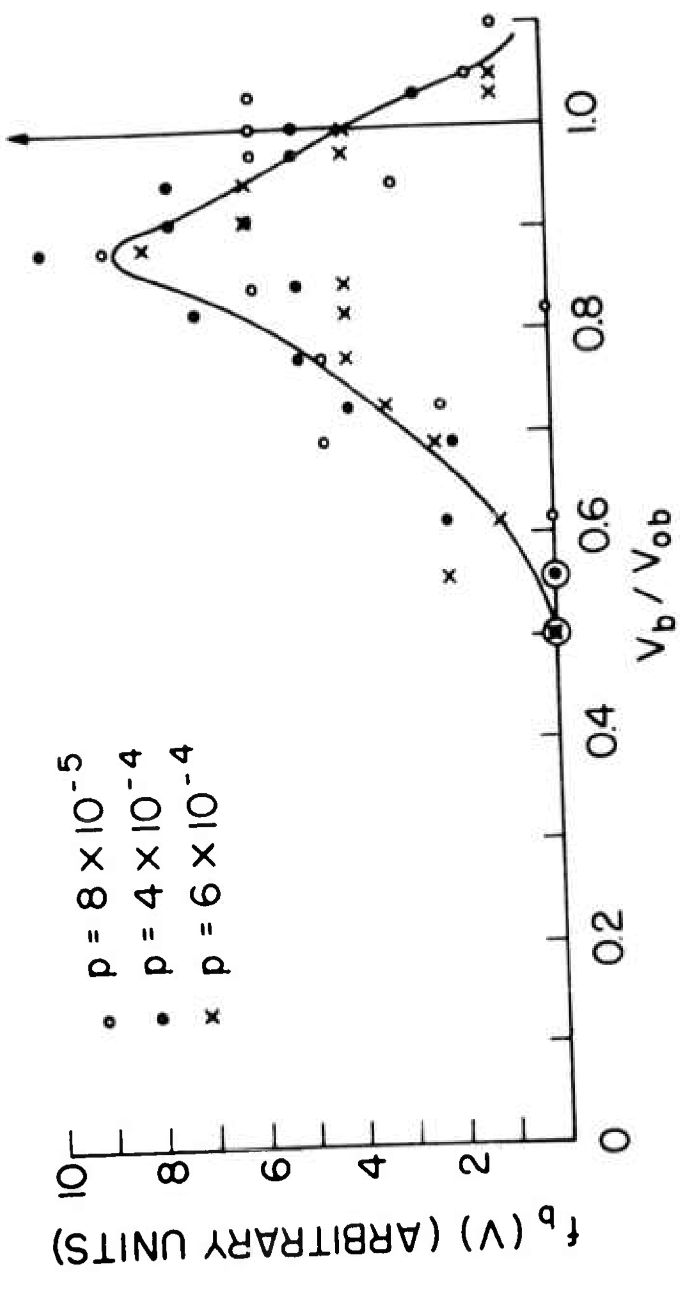


Figure 25. Collected Beam Velocity Distribution

$$E(z,t) = \frac{-Q}{\epsilon_0}\sqrt{1 + \nu^2/4\omega_0^2}\, \exp\left[-\frac{\nu}{2}(t - t_1(z))\right]$$

$$\cos\left\{\omega_0\left[t - t_1(z)\right] - \tan^{-1}\frac{\nu}{2\omega_0}\right\} u_{-1}\left[t - t_1(z)\right], \qquad (1)$$

where $\omega_0 = \omega_p\sqrt{1 - \nu^2/4\omega_p^2}$, u_{-1} is the unit step function, v = the electron-neutral collision frequency, and $t_1(z)$ = the time the sheet crosses the plane z. Since the plasma is assumed linear, the wakes can be superposed. The first computer experiments, ignoring collisions, showed unreasonably large beam-velocity spreads and fields for which the plasma oscillations were non-linear. Introducing a large collision frequency ($\nu = 0.2\omega_0$) reduced the magnitude of the electric fields. The results are shown in Figure 26. Even under these conditions the fields (Figure 26) are large enough to cause a beam velocity spread far exceeding the experimentally observed one. The beam was 2°/0 velocity modulated at ω_0, at injection. A large charge bunch forms at the position (150) of initial beam overtaking that is synchronous with the traveling electric field in such a manner as to maximally decelerate the beam. Two waves are clearly seen beyond overtaking. One has a phase velocity about equal to that of the wave before overtaking. The other wave is synchronous with clumps of beam charge at low velocity, and is very effective at slowing these clumps until downstream the beam is highly dispersed in velocity.

If the plasma had a longitudinal plasma density gradient, the large beam charge bunch formed at the point of overtaking would drive the plasma at other than the local plasma frequency; i.e., off resonance. Hence the electric field there would be much less than in the uniform case. An inhomogeneous plasma would also be expected to disrupt the slow phase velocity wave that formed beyond overtaking and dispersed the beam in velocity. This wave formed because the wakes of clumps of beam charge that pass a plane at time $2\pi/\omega_p$ apart add coherently. If ω_p is a function of distance, this condition cannot be maintained over an appreciable distance, and the wave will be broken up.

ONE-DIMENSIONAL BEAM AND PLASMA WITH DENSITY GRADIENTS ALONG THE BEAM

We introduce a linear longitudinal plasma density gradient, described by

$$\omega_p^2(z) = \omega^2 + az. \qquad (2)$$

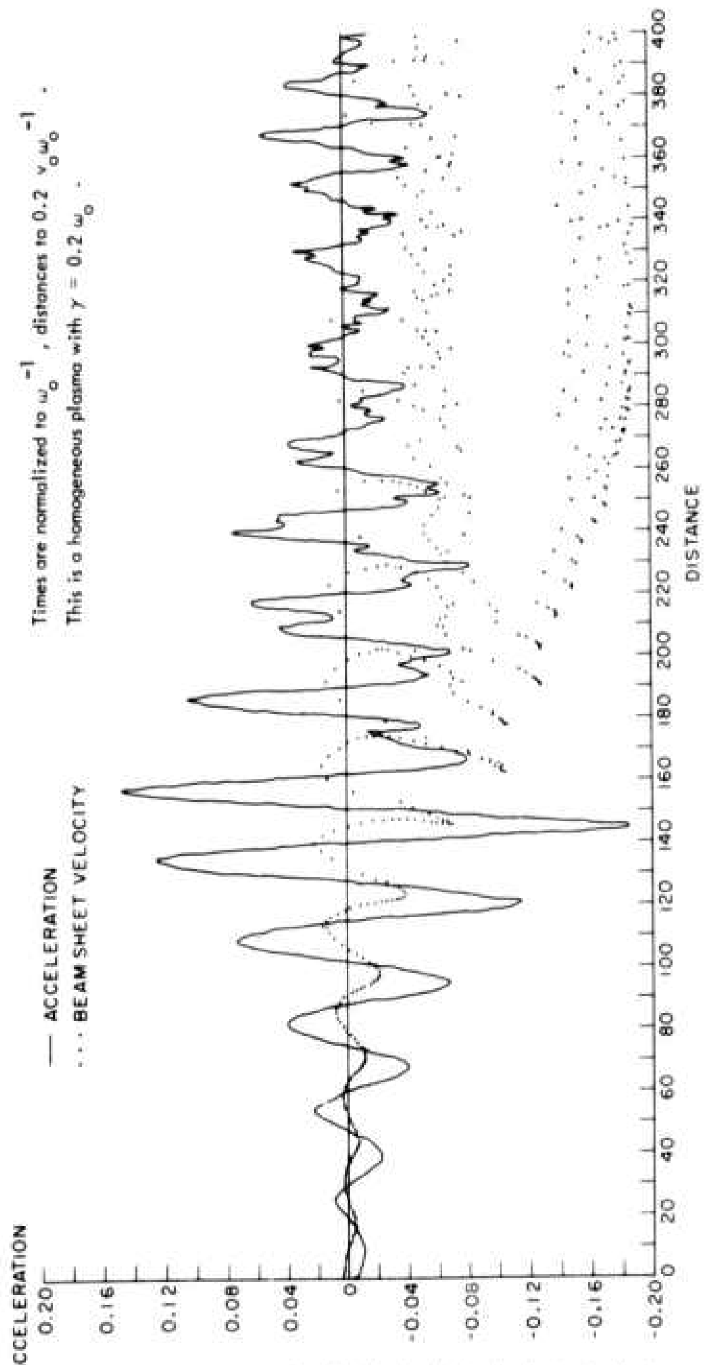


Figure 26. Snapshot of Sheet Velocity and Test Particle Acceleration vs Distance

The plasma is now assumed lossless ($\nu = 0$). "a" is chosen to be $\pi\omega_{p0}^2/L$, where ω_{p0} corresponds to the peak density of a sinusoidal density distribution in z. We assume a fixed frequency of excitation ω_{p0}^2, corresponding to the plasma frequency at the position of beam injection. The plasma is cold, lossless, linear and is treated analytically. The beam is 0.1% velocity modulated at ω. We have derived the linearized theory for a beam-plasma interaction with a gradient described by Eq. (2). We find that the first-order beam velocity grows as $I_0(Bz^{1/2})$ and the electric field is proportional to $I_1(Bz^{1/2})/z^{1/2}$, where I_0 and I_1 are modified Bessel functions, $B = 2\omega_{pb}/v_0 a^{1/2}$ and v_0 is the beam velocity. This linear theory is well matched by the results of the computer experiments shown in Figures 27 and 28, where the linear theory results are shown by the curves labeled v_{bIII}/v_{0b} and qE_{III}/m. We note that for a positive density gradient the variables can only grow beyond the plane where $\omega = \omega_p(z)$. Hence by the time overtaking occurs, the beam finds itself in a region where $\omega < \omega_p(z)$. The large beam charge formed at the plane of overtaking (Figures 27 and 28, in the vicinity of 150 distance units) drives the plasma off resonance, so the response is quite finite.

In these results the electric fields remain quite finite, but the assumption of plasma linearity is only marginally satisfied ($v_p/v_0 \approx 0.18$), and the beam velocity spread still exceeds that of the experiment.

FINITE DIAMETER BEAM AND ONE-DIMENSIONAL PLASMA WITH DENSITY GRADIENT

To obtain better quantitative agreement with experiment, we introduce disks to represent the beam. The fields generated by a disk moving through a plasma have been found for a uniform plasma and a constant velocity disk. There are two fields, a wake field and a nonoscillatory field. The acceleration on another disk due to the wake field is:

$$a(z,t) = \frac{-Q^2}{M\epsilon_0}\left[1 - 2I_1(\alpha)\,K_1(\alpha)\right]$$

$$\cos\left[\omega_p\left(t - \frac{z}{v_0}\right)\right] \quad u_{-1}\left(t - \frac{z}{v_0}\right) \tag{3}$$

where $\alpha = \omega_p b/v_0$, M is the disk surface mass (kg/m^2), Q is the disk surface charge, b is the disk radius, and K_1 is the modified Bessel

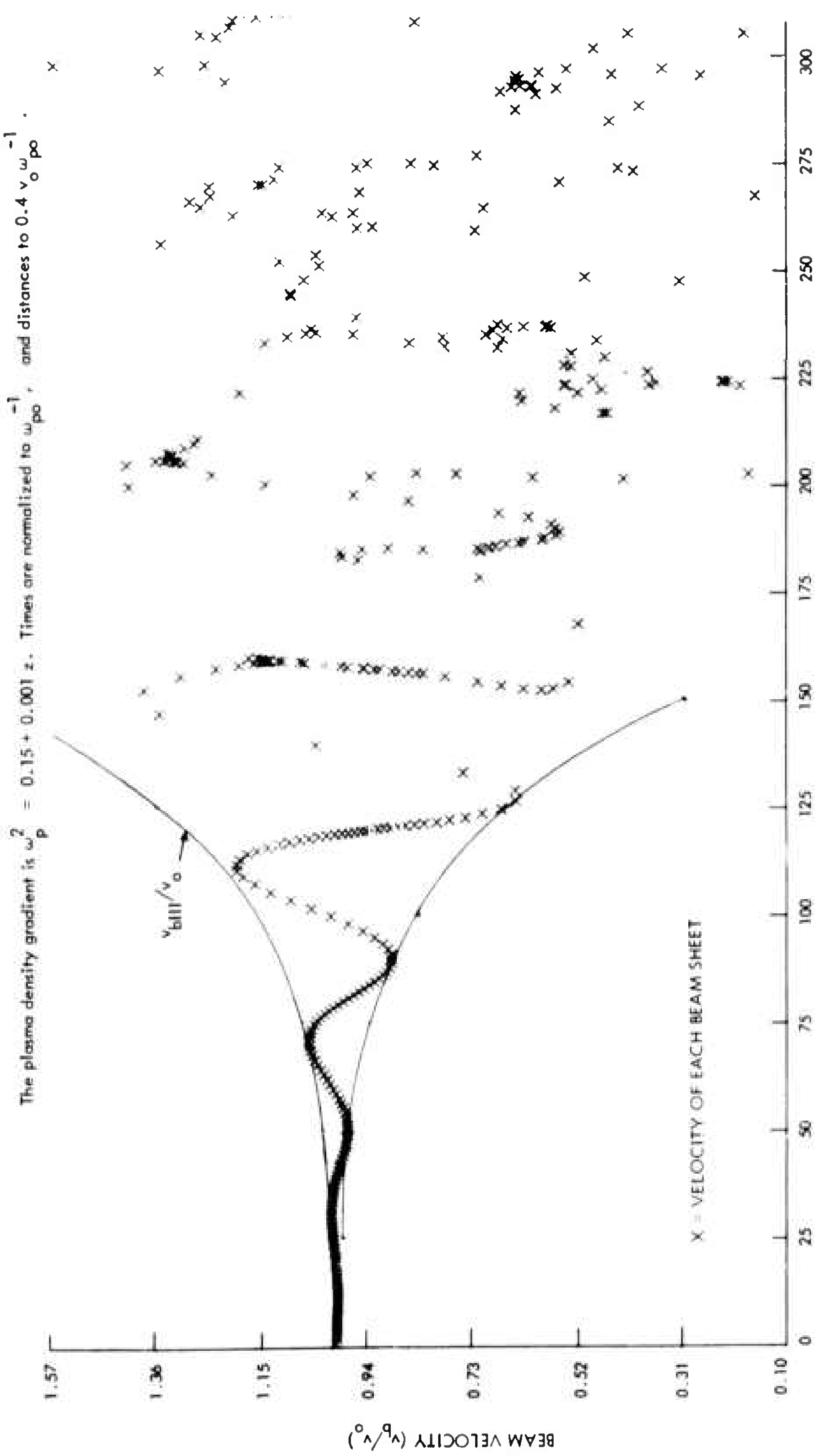


Figure 27. Snapshot of Beam Sheet Velocity

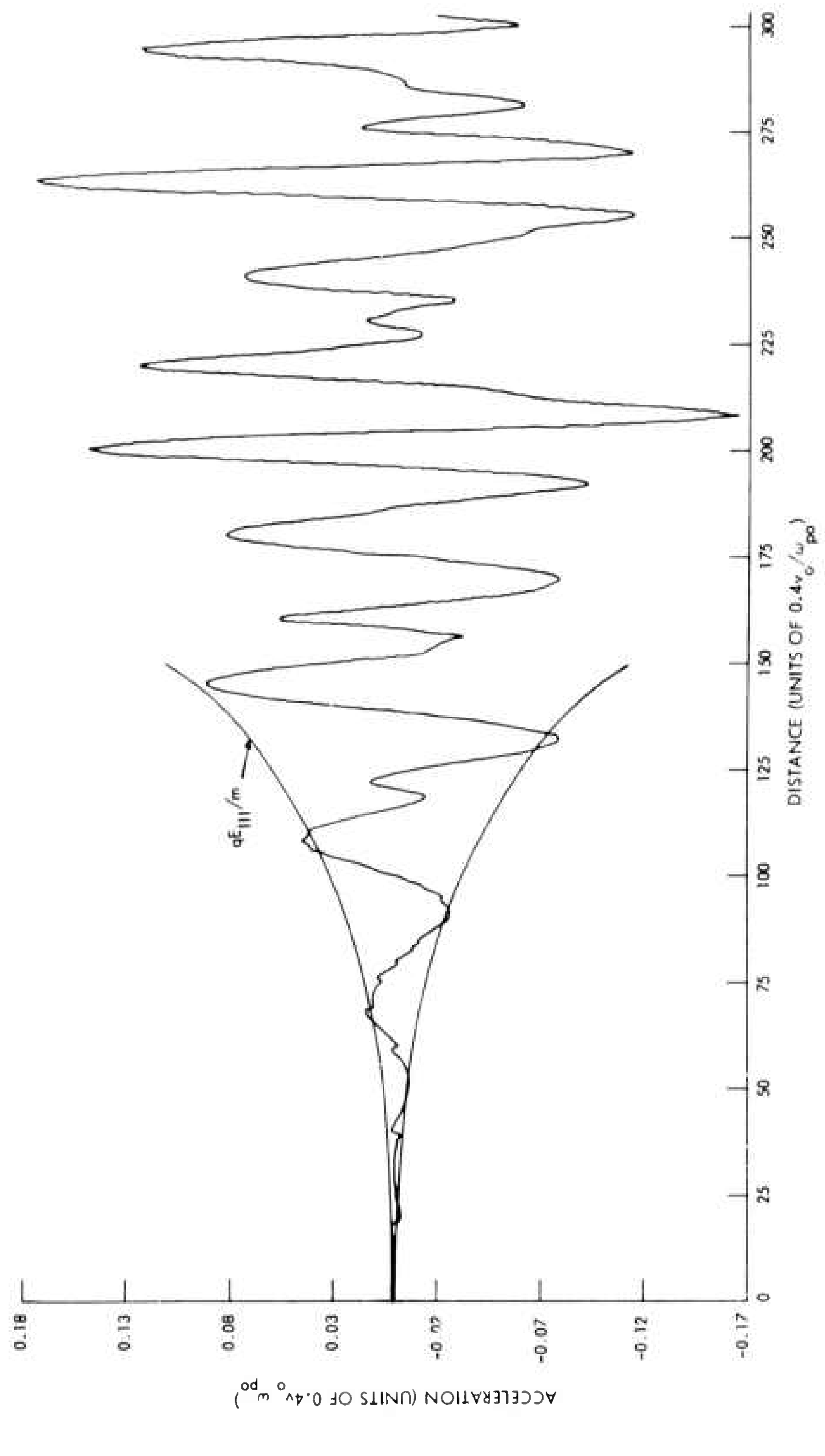


Figure 28. Snapshot of Test Particle Acceleration, with Plasma Density Gradient

function of second kind. The acceleration on another disk due to the non-oscillatory field ahead of the disk is:

$$a(z,t) = \frac{Q^2}{M\epsilon_0} \int_0^\infty \frac{dx\, x\, J_1^2(x)\, \exp\left[-x\Delta z/b\right]}{x^2 + \alpha^2}, \tag{4}$$

where Δz is the disk separation. The nonoscillatory field behind the disk is the mirror image of the field ahead.

We want to allow for variations in $\omega_p(z)$ and disk velocity. We do this approximately as follows. To calculate the wake field at a plane z we use Eq. (3), with the local $\omega_p(z)$ and the velocity the disk has when it passes z. To calculate the nonoscillatory field at a plane z, we use the local $\omega_p(z)$ and the present velocity of the disk. The computer results are shown in Figure 29.* The beam is velocity modulated at $z = 0$ over a frequency band extending from $(0.15)^{1/2}$ ω_{p0} to 0.5 ω_{p0}. This simulates the effects of plasma fluctuations of energy density $n\pi T$. The plasma frequency in this case varies spatially as

$$\omega_p^2(z) = \omega_{p0}^2 \sin(0.15 + az). \tag{5}$$

The plasma remains quite linear $(v_p/v_0 \approx \rho_p/\rho_{p0} \approx 0.03)$.

The velocity spread shown is comparable with experiment, but the interaction length shown is only 20% that of the experiment.

COMMENTS ON PROJECT MAC

Debugging and testing is performed much faster on time-sharing than on batch-processing. One evening spent on time-sharing often produces results that would take weeks otherwise. The use of disks to store programs and pseudo-tapes seems to the author to be much more reliable than using cards. Only very rarely have disk errors been detected, whereas cards are frequently mispunched or misread. An important

*In this calculation we let the region $0 < z < 200$ approach a steady state, collecting sheets at $z = 200$ before letting sheets excite fields beyond $z = 200$. Then we allowed sheets to reach $z = 250$ before collecting. After a steady state was approached again, sheets were allowed to reach $z = 300$. This was done to avoid the large transient fields that would be excited downstream by initial beam overtaking, which occurs far downstream early in the transient buildup.

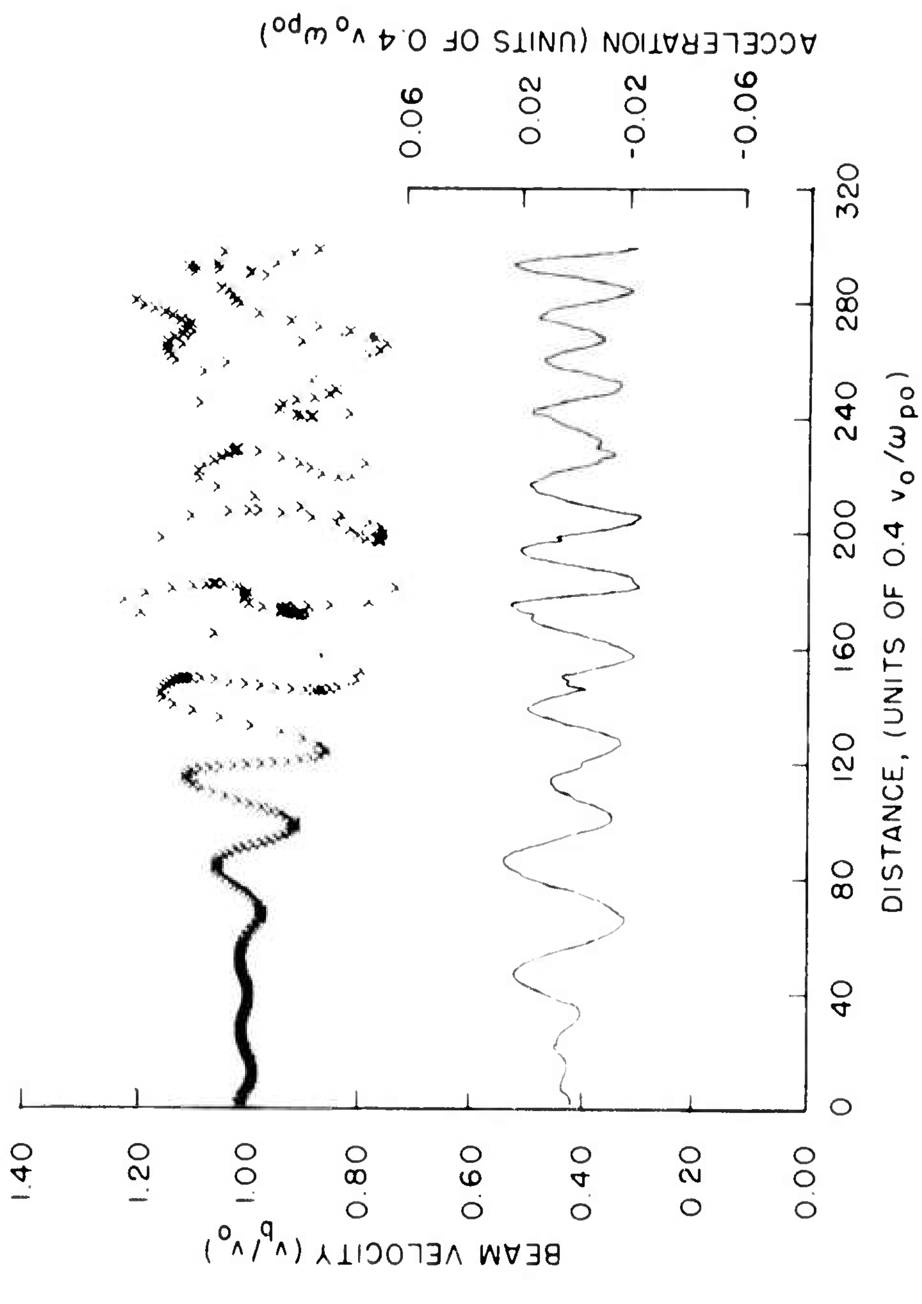


Figure 29. Snapshots of Beam Disk Velocity and Test Disk Acceleration vs. Distance, with Plasma Density Gradient

source of errors is eliminated since the input and output are handled only by the programmer.

A Solution of a Nonlinear Differential Equation - Herman M. Schneider

The time-sharing system of Project MAC was used to solve a non-linear differential equation in which one parameter was varied. The differential equation was

$$\frac{dx^2}{dt^2} = -\frac{1}{4}\,\omega_{p0}^{\;2}\left[3/2(x - x_0) + \frac{d}{\pi}\left(\sin\frac{2\pi x_0}{d} \sim \sin\frac{2\pi x}{d}\right) + \right.$$

$$\left. + \frac{d}{8\pi}\quad \sin\frac{4\pi x}{d} - \sin\frac{4\pi x_0}{d}\right] \qquad (1)$$

subject to the initial conditions $\frac{dx}{dt} = 0$ at $t = 0$, and $x = x_0 + \delta$ at $t = 0$. The quantities ω_{p0}, δ, and d are constants, and the solution is a function of t if a value of x_0 is chosen. Equation (1) was integrated numerically and the solutions for $x = x(x_0, t)$ were plotted versus t for various values of x_0. The purpose was to determine when $\frac{\partial x}{\partial x_0} = 0$, which can be seen on these plots by observing when two solutions for neighboring values of x_0 cross. By solving this equation on the Project MAC time-sharing system, the increment in x_0 could easily be chosen by observing the solution for the last value of x_0 used.

In Figure 30 the solutions of Eq. (1) are plotted for values of x_0/d between 0 and 1.1 in increments of 0.1.

This work was supported by the National Science Foundation under Grant GK-1165, and is available in the literature. (See Schneider, Appendix C.)

Numerical Solutions to Wave Propagation Problems - T. Kenneth Gustafson

During the past year we have been developing techniques for solving a particular type of partial differential equation. This project has been partially supported by the Joint Services Electronics Program under Contract DA-28-043-AMC-02536(E). This so-called parabolic equation is coupled to one or several other partial differential equations, involving only one such partial derivative in each. The particular system of interest is one in which a laser pulse enters a medium having some general properties which can be described in terms of differential equations – for

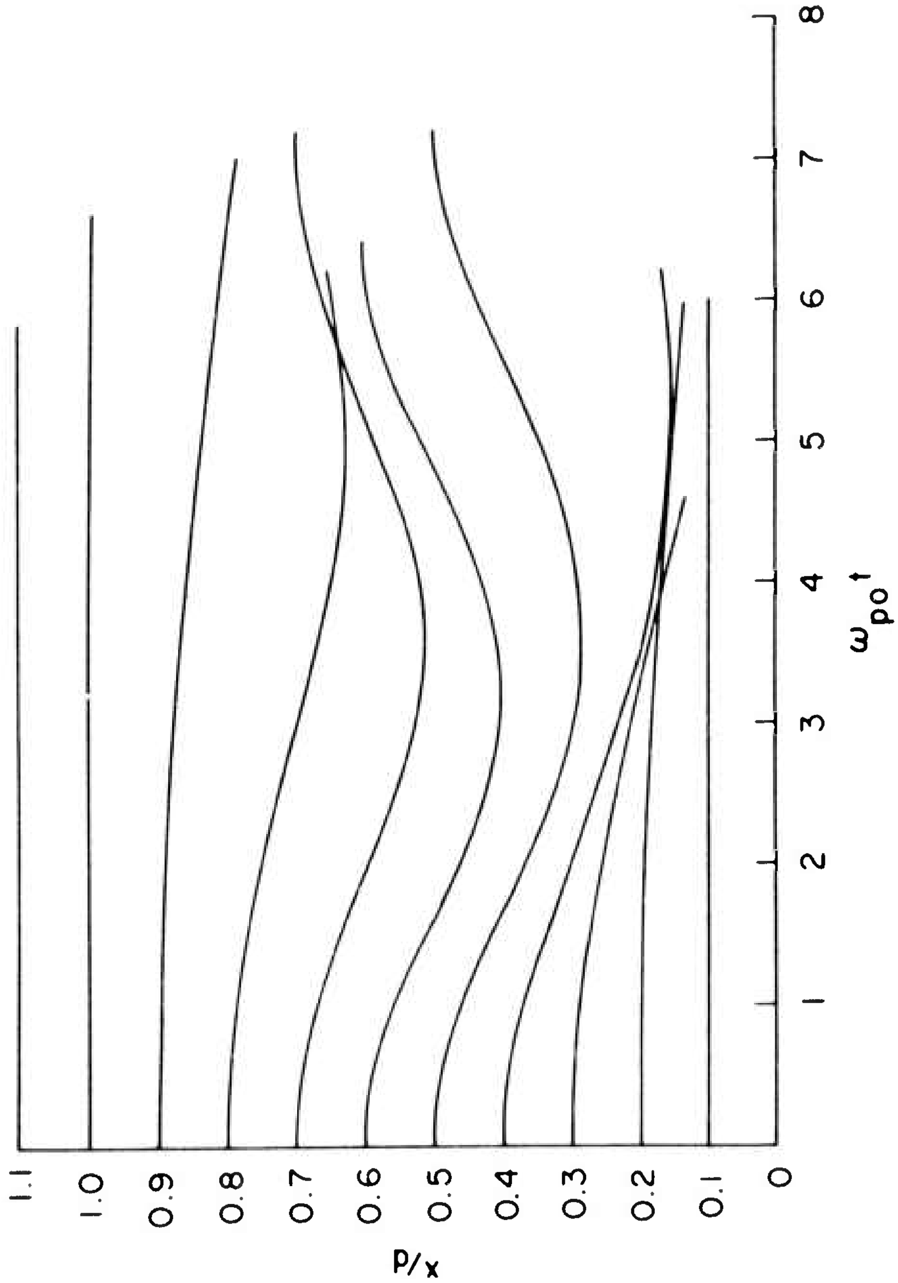


Figure 30. Plot of Values of x_0/d Between 0 and 1.1 in Increments of 0.1

example, a medium in which the population is inverted or one which exhibits some other non-linear index of refraction.

Some advantages of using the time-shared system were the readily available interpolation procedures and other subroutines, such as taking the Fourier transform. However, flexible parameter control was by far the most valuable advantage, when compared to batch-processing procedures. Indeed, it would have been impossible to attempt different approaches for solving equations of this type without such versatility and the ability to obtain nearly immediate response to changes. This was particularly true in the successful use of some integrating techniques, which up to this point, were justified on the basis of trial and error.

The procedure developed by us for solving ordinary differential equations is based upon the Runge-Kutta Technique. Previous applications of this technique cited in literature have appeared vague; also, no attempts to use it in solving coupled equations have been carried out.

Runge-Kutta is an averaging and smoothing technique which maximizes accuracy in a finite difference integration procedure. For ordinary differential equations, you can prove that step size can be made as small as desired. This, however, is a very time-consuming technique.

For partial differential equations, it is generally much more difficult to obtain an integration procedure that remains stable, that is, one for which an arbitrarily accurate numerical solution can be obtained. Even if a fairly accurate procedure is found for obtaining a specified step size in one of the independent variables, it is not necessarily true that when this step size is decreased that accuracy will be increased. If the returns were increased stability, one would be willing to sacrifice computation time.

By generalizing the Runge-Kutta procedure to one similar to the system described initially, we have obtained stability without incurring too much of an increase in computation time. The approach was to determine the direction of the linear characteristics of the parabolic equation and of the coupled "medium" equations. We then integrated directionally along these linearized characteristic directions, treating any non-linear terms as source terms of the system. In the integration procedure, moreover, we developed intuitively a Runge-Kutta type of averaging and smoothing technique for both spatial and temporal changes. This caused the resultant set of difference equations to be very tightly coupled amongst one another, which in turn seemed to imply inherent stability.

The integration procedure has been verified by comparison with a known solution for a specific system of equations. It has also been used since to investigate several propagation problems. Figure 31 illustrates the type of results obtained from a numerical study of the behavior of a laser pulse interacting with an amplifying medium.

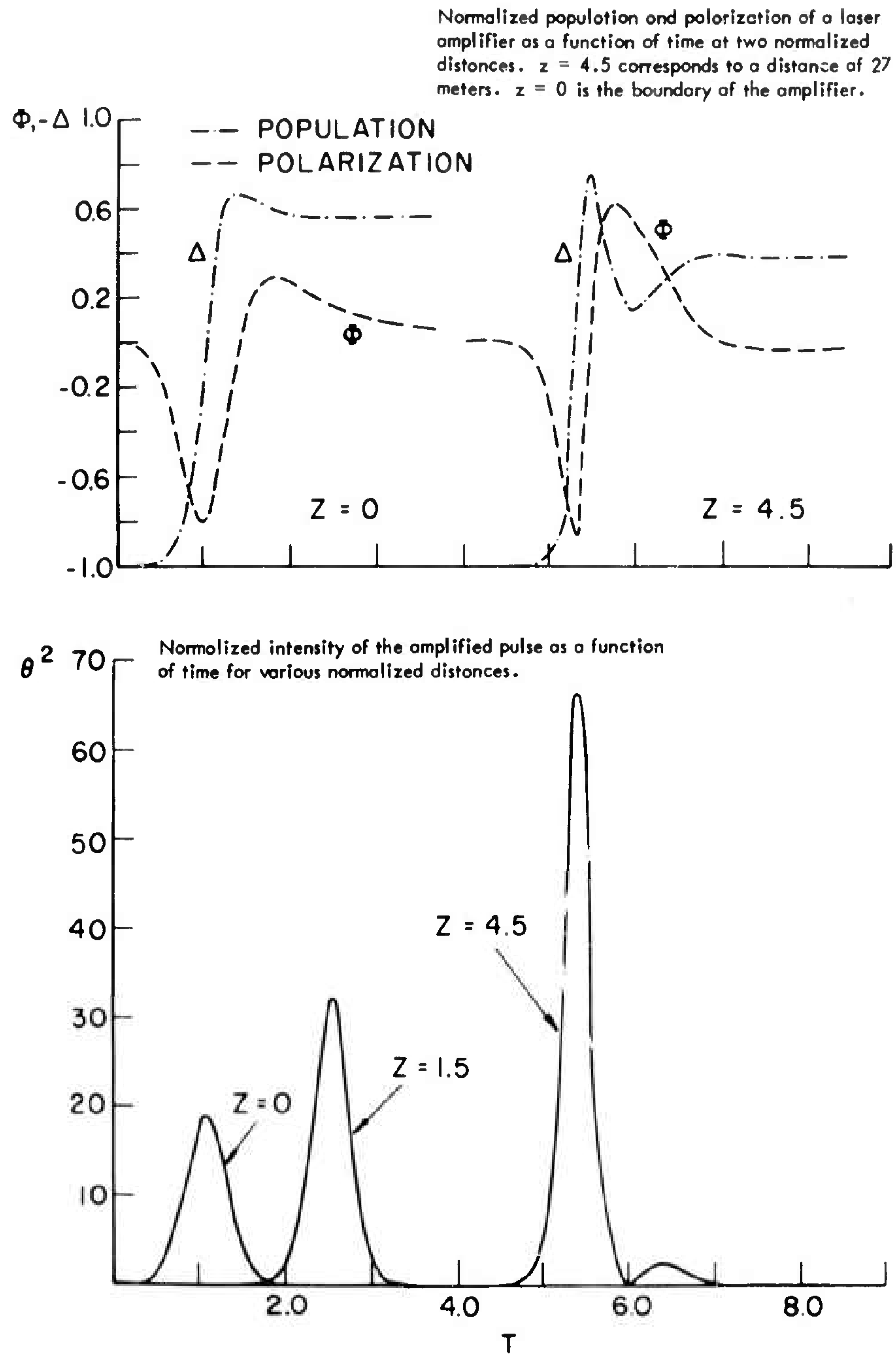


Figure 31. A Laser Pulse Interacting with an Amplifying Medium

It would be convenient if a procedure could be developed for integrating partial differential equations by a set routine. This does not appear too feasible. However, if the above approach is stable for more general equations, perhaps work could be carried out along this line.

At present we have begun to investigate the procedure mathematically and are attempting to determine error and stability criteria.

The Cauchy-Integral Root-Finding Method - Eleanor C. River and Elaine F. Brown

A method for finding the simple roots[1] of an arbitrary analytic function has been successfully implemented. The program (sponsored by the National Science Foundation under Grant GK-1165) is especially useful in plasma physics research, where exact solutions to dispersion relations are usually impossible, and numerical approximations are often valid only for a special type of equation. The method described below permits the user to input any function which is analytic in the domain of interest, except at a finite number of poles.

The problem is to find the roots of $f(z) = 0$, where $f(z)$ is a given complex function. A simple consequence of Cauchy's residue theorem permits us to state that a root, Z_o, of $f(z)$ can be expressed as

$$Z_o = \frac{1}{2\pi i} \oint_C \frac{zf'(z)}{f(z)}\, dz = \frac{1}{2\pi i} \oint_{f(C)} \frac{z}{f}\, df, \tag{1}$$

where C contains only one root. We are interested in the second form of this equation.[2]

Computation of Z_o, using (1), requires the following procedure:

1. Display $f(z)$ and determine the existence of a root by inspection.
2. Enclose the root by a curve, C.
3. Compute and display $f(C)$ in the $w = f(z)$ plane. If $f(C)$ does not encircle the origin or encircles it more than once (indicating the presence of additional roots), return to Eq. (1) and enter C again.
4. Using the Trapezoidal rule, evaluate the line integral

$$Z_o = \frac{1}{2\pi i} \oint_{f(C)} \frac{z}{f}\, df$$

This approximation to Z_o is used to obtain a refined value of Z_o, using for C a circle centered at Z_o, and repeating steps 2 to 4 until a sufficiently

precise value is determined. Previous work indicates that convergence is typically quite rapid.[1,2,3]

The programming effort required to implement this procedure has been minimized by using the extensive facilities provided by the TOCS, which was developed on CTSS and is available to all users. (See Winiecki, Appendix C.) Primarily intended as a high-level desk calculator for the computationally unsophisticated user, the TOCS makes available techniques of man-machine interaction which facilitate selection and display of curves for evaluating the line integral of Eq. (1).

Modeling of Speech Production - William L. Henke

Work continues on an articulatory level model of speech production. The evolution of such a model, implemented as a computer simulation, serves both as a stimulant for the generation of new ideas and theories about the speech production process and as an agent to clarify, codify, and test those theories. Such a model may also be useful as the core of a system of speech synthesis by machine. This project is being supported partially by the United States Air Force, under ESD Contract AF19(628)-5661 and by the National Institute of Health, Grant 5 R01 NB-04332-04.

The present model accepts an input of discrete phonetic segments (limited to vowels and stops) and generates a continuous description of the time-varying state of the vocal tract (midsagittal section). Each input segment causes some members of a finite set of generative attributes or "instructions" to be invoked. The principal attributes are configurative (generalized targets with both shape and position data) and manner attributes. Invoked attributes are changed discretely in time — there is no blending of attributes — and pre- and post-coarticulation are realized by anticipation and noninstantaneous response, respectively.

---

1. For an extension of this method to poles and non-simple zeros, see McCune, J. E. and Fried, B. D., The Cauchy-Integral Root-Finding Method and the Plasma 'Loss-Cone' Instability, CSR TR-67-1, Center for Space Research, M.I.T., January 1967.
2. This method was proposed as a computational tool by McCune, J. E., "Exact Inversion of Dispersion Relations," The Physics of Fluids, Vol. 9, No. 10, pp. 2082-2084, Oct. 1966.
3. Fried, B. D., On-Line Root Finding in the Complex Plane, Report No. 9990-7308-R0000, Physical Research Center, TRW Systems, Redondo Beach, California, July 1966. Fried implemented this root-find method using the Culler On-Line Systems from which the TOCS is modeled.

The graphical input/output facilities and the cineradiographic data input and retrieval techniques (using the ESL Display Console) which were discussed in the previous progress report have been augmented and improved.

In addition to the model itself, work is underway on programs to relate articulatory descriptions to acoustic features of the speech signal. The following sequence is now operational:

1. Graphical input of a midsagittal section description of the vocal tract.
2. Calculation of the intersections of this description with a system of transverse planes.
3. Calculation of a so-called area function from the transverse plane description.
4. Calculation of the magnitude and phase spectra of the vocal tract transfer function.
5. Calculation of natural frequencies (formants).

This acoustic level system will be merged with the articulatory model so that formant tracks can be generated by the model for comparison with spectrographic data. A time-domain digital simulation of the acoustic wave generation is envisioned for eventual speech synthesis, but this will first require interconnection of the Speech Group small computer (a PDP-9) with the MAC system.

As a sideline, computer-generated movies (whose production was based on knowledge acquired during research with the articulatory model) were used to produce some educational films for an introductory electrical engineering subject.

The Psychophysical Reality of the Distinctive Features of Phonology - Dennis H. Klatt

A number of feature systems have been proposed to categorize the phonemic distinctions made in a language. Recently, Wickelgren[1] conducted a series of short-term memory experiments wherein errors in recall of English phonemes were related to three distinctive feature systems. A procedure for predicting the rank-order of confusions between pairs of stimuli was used to evaluate the three feature systems. This work was partially supported by the United States Air Force ESD,

1. Wickelgren, W.A., "Distinctive Features and Errors in Short-Term Memory for English Consonants", J. Acoust. Soc. Am., Vol. 39, pp. 388-398 (1966)

Contract AF19(628)-5661 and the National Institute of Health, Grant 5 R01 NB-04332-04.

The data of Wickelgren, in confusion matrix form, have been re-analyzed within a framework and a set of assumptions that permit the use of metric statistics to obtain an optimum set of binary distinctive features. The procedure is to divide the phonemes into two sets such that members of the same set are often confused, and two phonemes not in the same set are not confused as often. A random variable has been defined and maximized by reassigning phonemes to the sets in the manner of a "hill-climb". Further optima are found under the conditional assumption that the previously found optima are present.

Preliminary results indicate that several binary distinctive features appear as optimal sets. These optimal sets have been given the names: 1) hissing and hushing, 2) sonorant, 3) constricted, 4) voiced, and 5) anterior. The implications of these results will be described in a manuscript now in preparation.

Computer Aids for Linguists - G. Hubert Matthews

The goal of the initial phase of the programming is to provide a tool for the linguist developing an explicit transformational grammar. Work in this area is being partially supported by United States Air Force ESD, under Contract AF19(628)-2487. Basically, the process involves the linguist iteratively doing the following three kinds of activity:

1. Defining his grammar

The linguist uses the normal CTSS editing functions (EDL or QED) to state his rules as arguments to the LISP function DEFTRAN, and to set certain parameters indicating the order in which his rules are to apply - it is assumed that they are cyclic. For example, if the linguist has the transformation which he would usually write as (1), below, he types it as (2).

It-Repl

	X	NP	it	S	for	S	NP	VP	VP
(1)	1	2	3	4	5	6	7	8	9

substitute 7 for 3 and erase 7

Chomsky-adjoin 4 to the left of 9 and erase 4

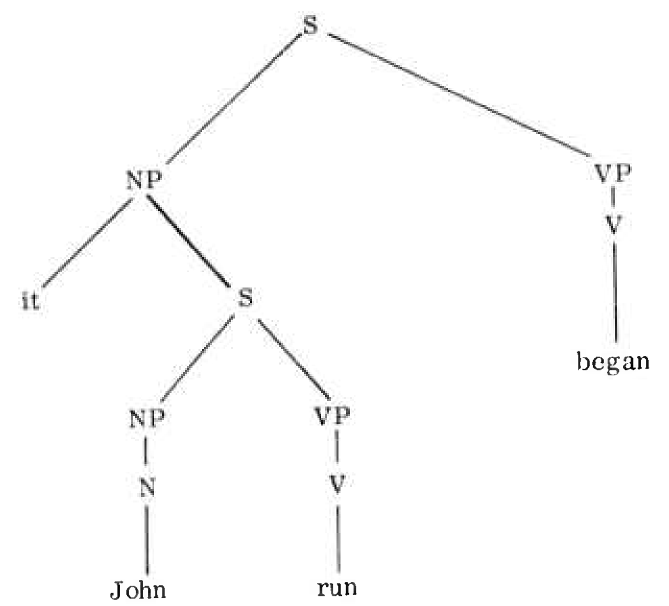
```
deftran ( ( itrepl
( x ( it ( for ( np vp ) ) ) vp )
```

```
(2) (   np   s     s )
    sb  ( -7 3 )
    cadjl  ( -4 9 )
                   ) )
```

2. <u>Constructing base trees to which the rules are to be applied</u>

The linguist has available a series of functions for inputting and for gradually modifying trees. The trees are printed by the typewriter in a 'sideways' format, with each node numbered. For example, the typical tree of (3) is printed as (4).

(3)




```
(4)    1S     2NP     3IT
                      4S     5NP    6N     7JOHN
                             8VP    9V     10RUN
              11VP    12V    13BEGAN
```

In looking at this form, the linguist may modify it by using any of a number of LISP functions which take as arguments the location of the place in the tree to be changed and the change to be made. For example, to change the predicate of the embedded sentence from 'run' to the verb phrase 'like linguistics', the linguist could type as at (5), and this would change (4) into (6).

(5) chdat (8 (v (like) np (n (linguistics))))

```
     1S      2NP      3IT
                      4S      5NP      6N      7JOHN
(6)                           8VP      9V      10LIKE
                                       11NP    12N      13LINGUISTICS
             14VP     15V     16BEGAN
```

It is also possible to predefine 'shells' of often used subtrees and then put these where desired into the tree being built. While the most desirable way of building trees would be to use a scope and lightpen, it is hoped that the system in use here will still be found convenient.

3. Testing the grammar

Having specified a grammar (i.e., a set of ordered transformations) and a tree, the linguist is now ready to test the grammar by applying the transformation to the tree. He may elect to be shown the tree after each transformation has changed it, or he may have the system tell him only which transformations have applied and, after the derivation is finished, he may ask to see the trees that resulted from just those transformations he is interested in. If any transformation is optional and can be applied to the tree, the linguist is asked whether or not he wants it to be applied. At this point he may ask to look at parts of the tree and to find out how it will change the tree if it does apply. When he discovers an error (i.e., a case where a transformation does not do to the tree what he had expected), he may then either change the tree or modify the transformation and try again.

The programs described above are running, and a number of linguists have expressed an interest in using them. It is planned to make additions and modifications to the programs as experience indicates that they are useful. It is also planned to modify the program as it now exists so that transformations using grammatical features will be accepted.

Current Distributions on Equiangular Spiral Planes - Gary D. Bernard

The exact Fourier-Bessel integral forms of current distributions for both a dipole-excited spiral plane, and for a dipole-excited spiral plane that is spaced above a perfectly conducting plane were evaluated, under contractual support from the Joint Services Electronics Program, Contract DA-28-043-AMC-02536(E). Ordinary and modified saddle-point approximations, including up to three terms of the asymptotic series were used. Using Project MAC facilities, numerical comparisons were made of surface wave and space wave terms in the ordinary approximation, and of complementary error function terms and terms of the

asymptotic series in the modified approximation. Special attention was paid to establishing parameter ranges over which a particular term was important, and to comparing ordinary and modified approximations. Results of this study were reported at the 1966 U.R.S.I. meeting. (See Bernard, Appendix C.)

A Sinusoidally Stratified Model of the Insect Corneal Nipple Array - Gary D. Bernard

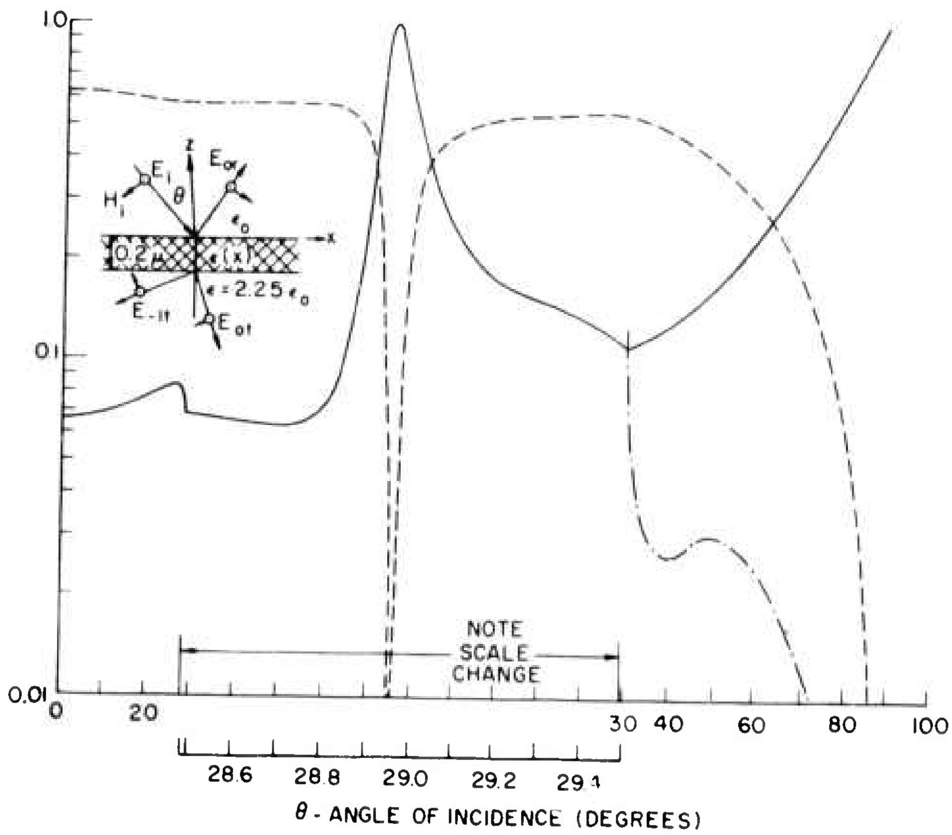
Research work has been conducted on insect corneal nipple array with support from the Joint Services Electronics Program, Contract DA-28-043-AMC-02536(E). A simplified mathematical model of a corneal surface was constructed and studied. The corneas of many insects are completely covered with a hexagonal array of nipple-like structures[1] with characteristic dimensions in the order of a half-wavelength of visible light. The model studied was of a TE plane wave incident upon a dielectric half-space covered by a sinusoidally stratified dielectric layer. The dielectric constant of this layer varies sinusoidally in the one direction that is both parallel to the interface and in the plane of incidence, but remains constant in the other two mutually perpendicular directions.

The Project MAC CTSS facilities were used to study zeroeth and minus-first reflected and transmitted waves as a function of incident angle and wavelength for sets of parameters corresponding to both subsurface nipples and nipples-in-air. Besides giving us a feeling for relative strengths of these waves, the numerical study indicated the possibility of scattering resonances, i.e., situations where the incident wave is totally reflected from the layer. This happens in the subsurface case when the average dielectric constant of the layer is greater than that of the dielectric half-space, and when the layer is thick enough to support a surface wave. Such a situation is shown in Figure 32, for which total scattering occurs at an incident angle of 28.96° with a full width in transmission coefficient of about 0.1°.

The on-line feature of CTSS was a great help in learning how the scattering resonance behaved with changes in parameters of the layer, and in showing the resonance to be related to a surface wave propagating in the slab. For instance, as the slab thickness is reduced (Figure 32) the angle for total reflection approaches 30° (the angle at which the transverse propagation constant in the dielectric half-space of the minus-first diffracted order is zero) while the null width becomes very narrow. When

---

[1] W. H. Miller, A. R. Møller and C. G. Bernhard, "The Corneal Nipple Array", The Functional Organization of the Compound Eye, Pergamon, 1966, pp. 21-33



NOTE $\epsilon(x) = 2.56\,\epsilon_0\,(1 + 0.1211 \cos 5x)$ where x has units of microns, and for wavelength of incident wave of 0.4μ

———————— is $\left|\frac{E_{ar}}{E_i}\right|^2$ for zeroeth order reflected wave

- - - - - - - - - - - is $\left|\frac{E_{ot}}{E_i}\right|^2$ for zeroeth order transmitted wave

—·—·—·—·—·— is $\left|\frac{E_{-1t}}{E_i}\right|^2$ for minus-first order transmitted diffracted wave

Figure 32. The Diffraction and Scattering from a Sinusoidally Stratified Dielectric Layer

the thickness decreases to a critical value, the scattering resonance is no longer present: this critical thickness is at surface wave cutoff, i.e., the transverse attenuation constant in the dielectric half-space of this surface wave is zero.

It was quite convenient to track this very sharp resonance to extinction using a trial-and-error procedure while on-line at a CTSS console. The on-line CTSS system takes maximum advantage of the learning capabilities of the operator, making it clearly superior to a batch-processing system. Also, the computer program can be much less complicated than would be required for the batch-processing approach.

We don't know whether such scattering resonances actually occur in insect corneas. It is possible that the average dielectric constant of the subsurface nipple layer is too low, or that corneal curvature causes severe broadening and lowering of the resonance peak. Effects of curvature in the theoretical model are currently being considered, and a search is continuing for evidence of scattering resonances in insect corneas.

<u>Blood Cell Classification</u> - William L. Black, Ian T. Young

The objective of this project, started in March 1967, is to classify white blood cells automatically, recognizing pathologies when they occur. Work has been supported principally by the National Institute of Health, Grant 1-P01-GM-14940-01, and by the Joint Services Electronics Program under Contract DA-28-043-AMC-02536(E).

Initial investigations dealt with the logistic problems involved in converting photomicrographs to stored digital data and with evolving a library of routines for handling, processing, and displaying pictures. We are now working on ways to expedite the recognition task through experimentation on the pictures with various algorithms.

The immediate problem is to separate the cells from the background, a process complicated by a serious non-uniformity of the background density. Though initial attempts at high-pass filtering were not effective, they did suggest other procedures which we are now trying.

Among our utility programs are two which might be useful to other programmers. One produces histograms of picture brightness, while the other prints out a picture on the off-line printer with a variance of 17 gray levels.

<u>Touching and Overlapping Chromosomes</u> - Oleh J. Tretiak

This project, which is to develop algorithms for karyotyping chromosome spreads in which the chromosomes are touching or overlapping, is supported principally by the National Institute of Health, Grant

1-P01-GM-14940-01, and by the Joint Services Electronics Program, Contract DA-28-043-AMC-02536(E). The implementation of such a system is shown in Figure 33. The approach is one of analysis by synthesis. The data is abstracted from the image, and hypotheses about the image constitution are formed on the basis of this data. The plausibility of these hypotheses is checked against images encountered, and any case of ambiguity is resolved by abstracting further data from the image.

At present, only the right-hand block of the system, the reduction process, has been analyzed and is shown in Figure 34. An image, scanned over a 256 by 256 grid, is first separated into manageable segments by isolating regions included in contours traced at a fairly low density level. Each region contains one or more chromosomes. The data contained in some of the regions is shown in Figure 35. Figure 35a contains an A group chromosome, Figure 35b an F group chromosome, and Figure 35d an E group chromosome. The region in Figure 35c contains chromosomes from the C, E, and F groups. The lines in Figure 35 are contour traces at the density level used to identify the regions and at three other equally spaced density levels. It is evident that the density of chromosome images is quite variable, and that isodensity traces do not correspond to the shapes ascribed to them by biological models.

The second step of the reduction process attempts to locate the probable positions of the chromatids. Smoothed first and second derivatives are observed and the locations of points with substantial second derivative and of small first derivative are noted. The regions found by this process are shown in Figure 36. These regions are examined in detail by a "bug" which follows ridges, and the tracks of this bug are the probable chromatid locations. These "bug tracks" are shown in Figure 37.

Programs for the above operations have been written in MAD. The images in Figures 35, 36, and 37 were photographed from the ESL Display.

Private Libraries for TIP - E. C. River, M. M. Pennel, and H. S. Davis

In the relatively new field of plasma physics, the pertinent literature consists almost entirely of journal articles and special technical reports, most of which are not available in the M.I.T. libraries; consequently, researchers have found it essential to build up their own private libraries. Incorporation of the contents of these many libraries into one body of material which could be searched by TIP[1] would permit colleagues in the

[1] W. D. Mathews, "Searching TIP Bibliographic Files," The Compatible Time-Sharing System: A Programmer's Guide", P. A. Crisman ed., second edition, The M.I.T. Press, Cambridge, Mass., August 1965 (Library of Congress No. 65-25206), Section AH.2.16.

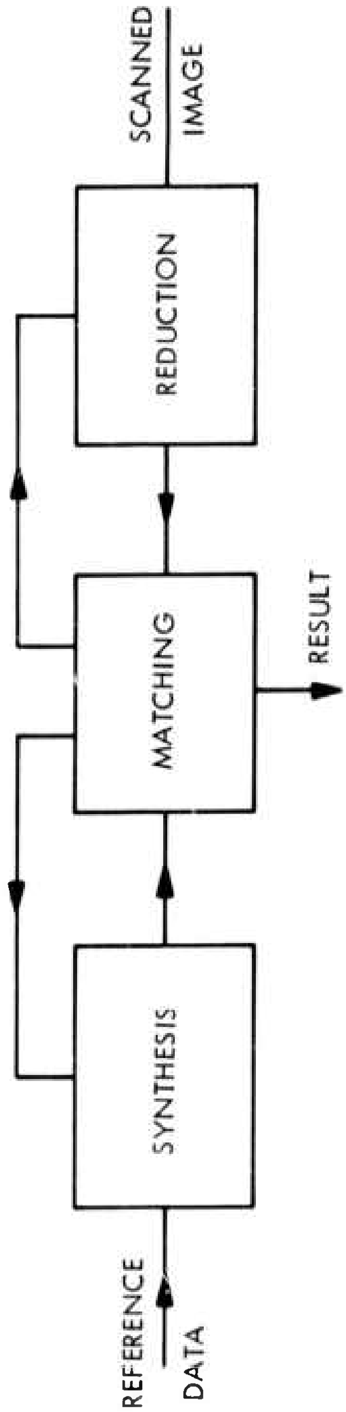


Figure 33. A Model of the Recognition Program

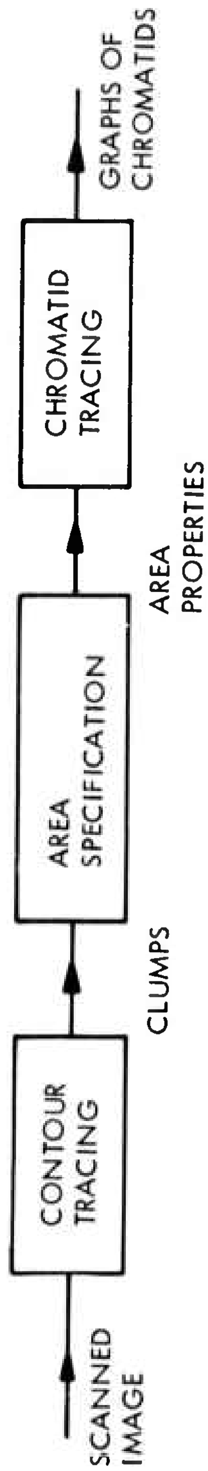


Figure 34. Reduction Process

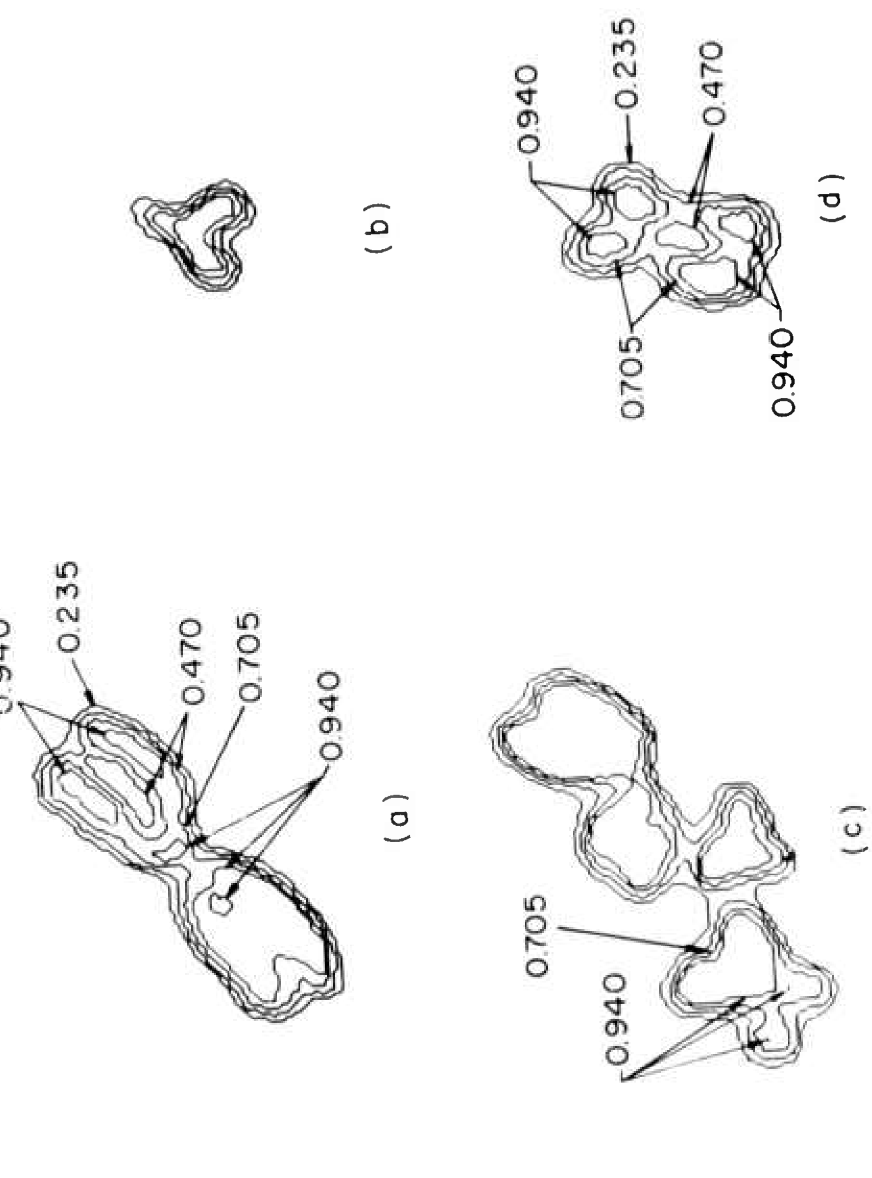


Figure 35. Data from Chromosome Regions

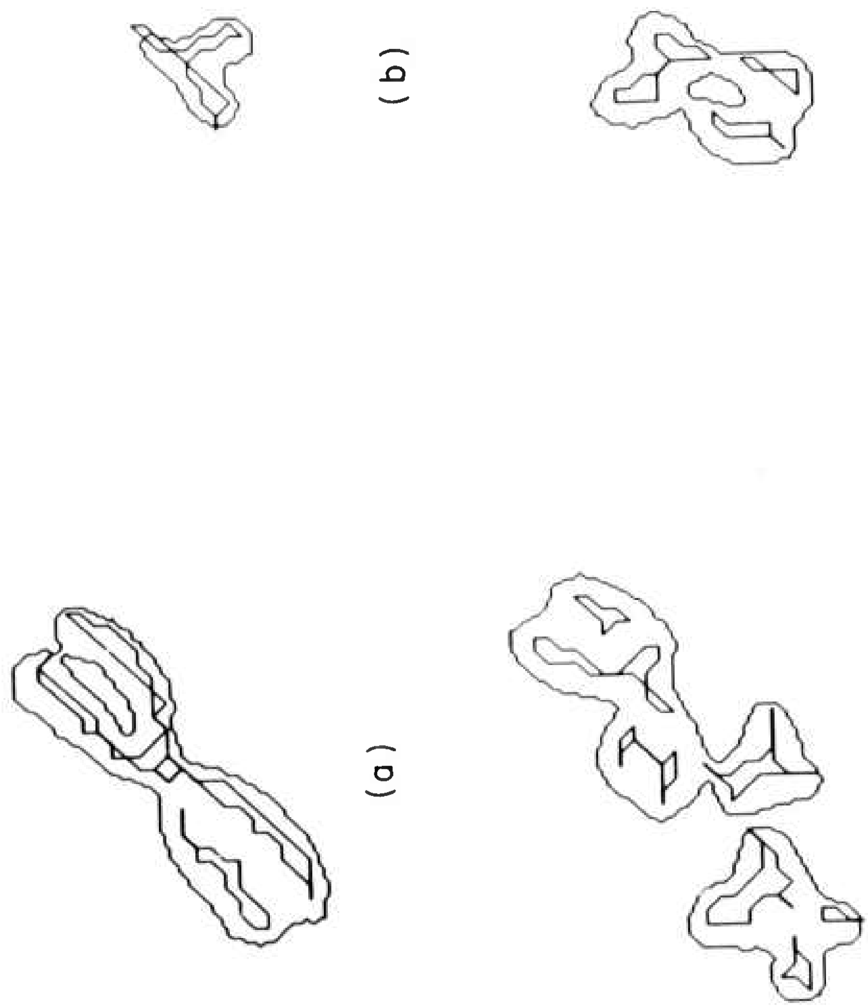


Figure 36. Location Points of First and Second Derivatives

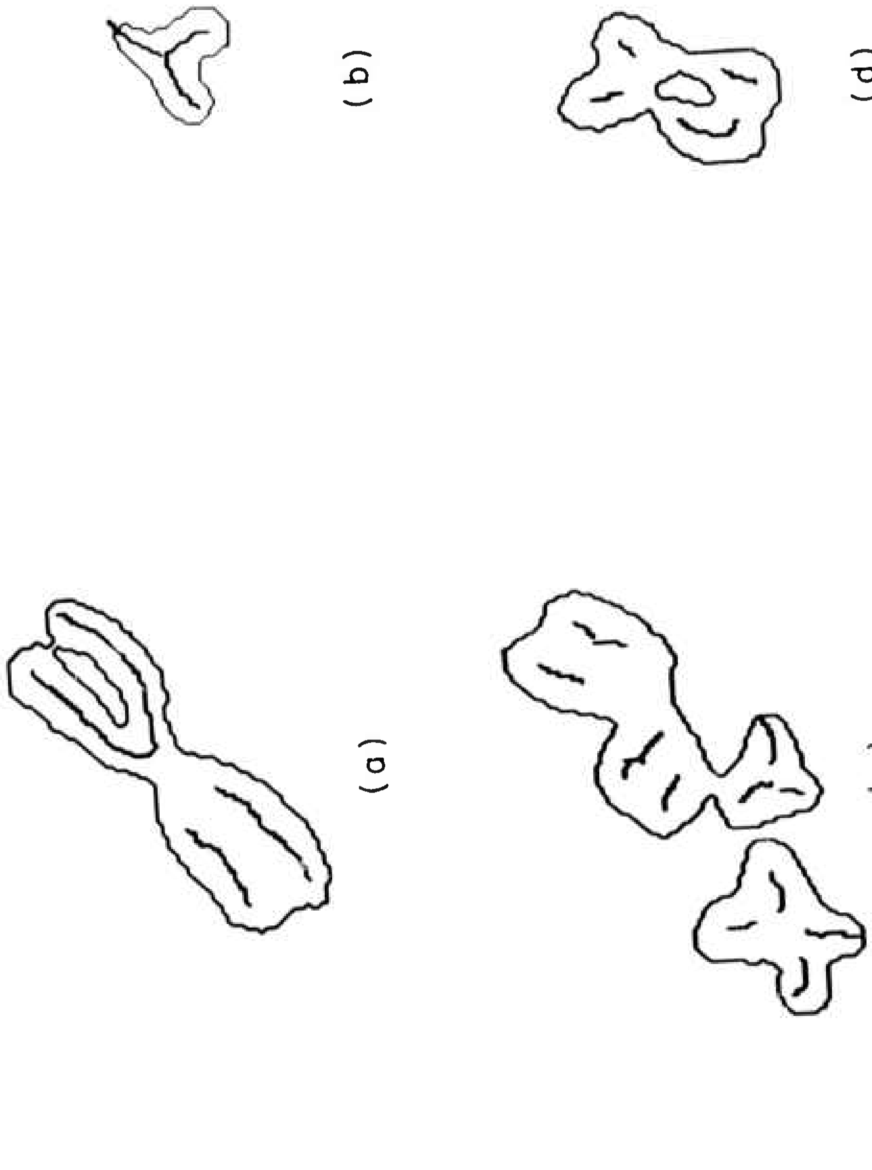


Figure 37. Tracks of Probable Chromatid Locations

field to avoid unnecessary duplication of expensive or scarce materials by giving them ready access to libraries other than their own. A plan for such a project has evolved, with partial support from the United States Atomic Energy Commission under Contract AT(30-1)-1842.

During the past year, work was begun in conjunction with the R. L. E. Document Room to create a group of private library files which can be searched by TIP. These files serve also as catalogues to the contents of the libraries, which now include those of Professors Allis, Bekefi, Bers and Brown, at whose suggestion the project was undertaken. It is possible to search the files singly or together in any combination.

Although the cataloguing of private libraries is a formidable clerical task, the file creation and editing facilities of CTSS have relieved secretaries of most of the tedious details of maintaining and updating active files. In fact, the major effort involved in this project was the actual manual sorting of the materials in each library and the decisions about which should be catalogued. By no means a full-time project, there are at present 107 entries in Professor Allis' file, 92 entries in Professor Bekefi's file, 336 entries in Professor Bers' file, and 608 entries in Professor Brown's file.

Programming Support and Development - Eleanor C. River and Martha M. Pennell

R. L. E. faculty members, students, and DSR Staff have made extensive use of facilities provided by Project MAC. Their activities have been aided by the R. L.E. Computation Group which has guided users by providing program development, consulting services and instruction on the use of CTSS. Typical problems handled by the group range from short numerical calculations to the use and maintenance of special-purpose subsystems to the creation of special libraries for TIP users. This work has been sponsored by Joint Services Electronics Program under Contract DA-28-043-AMC-02536(E).

Interest in CTSS is continually growing: many demonstrations have been given, some as tutorial sessions for individual user groups at their own consoles, others as two-hour seminars on CTSS usage. Secretaries were instructed in the use of TIP to enable them to create private libraries for TIP searches. An investigation of MAP and a subsequent demonstration to interested graduate students took place in the fall of 1966. The TOCS has been a great aid in implementing the Cauchy-integral root-finding method; we plan to introduce other members of R. L. E. to the TOCS in the near future. As a result of these demonstrations, more individuals within the laboratory have become aware of the possibilities and capabilities of CTSS and have begun to use the system.

The plasma physics groups have been heavy users of CTSS, and the R. L. E. Computation Group has assisted several graduate students in the development of specialized subsystems. One of these, FOLLOW, written by M. G. Smith (see Smith, Appendix B), is a program which automatically locates the multiple zeroes of a polynomial plasma dispersion function and plots the loci of these roots as functions of various physical parameters. FOLLOW relies heavily on the display facilities available at the ESP Display Console, as well as on the obvious interactive capabilities provided by CTSS.

SCHOOL OF ENGINEERING

Project Intrex

A. The Augmented Catalog
B. Text Access

The MAP System

The TREC System

Chempos: A Chemical Engineering Language

Optimization Methods Applied to Ship Design

The Computerization of Series 60 Resistance and Propulsion Experiments

Toward General Stress-Strain Analysis

Equilibrium Problem Solver (EPS)

Solution of Viscous Fluid Flow Problems

Two-Dimensional Stress Analysis

ENPORT: A Multiport System Simulator

Modeling and Analysis of Multiport Systems

Identification of Non-Linear, Lumped-Parameter Models for Dynamic Systems

Academic Staff

C. A. Berg
J. W. Brackett
S. A. Coons
K. F. Hansen
R. Kaplow
P. Kugel
J. C. R. Licklider
T. Lin
P. Mandel
F. A. McClintock
H. S. Mickley
C. F. J. Overhage
R. P. Parmelee
H. M. Paynter
F. J. Reintjes
R. C. Rosenberg
T. B. Sheriden

Non-Academic Research Staff

G. L. Benton
B. F. Boudreau
C. Garman
H. F. Gronemann
R. L. Kusik
R. Leopold
J. B. Lovins
R. S. Marcus
R. L. Stetson

Research Assistants and other Students

R. E. Banks
P. Choong
C. Chryssostomidis
D. L. Flamm
J. C. Free
R. P. Greer
R. Haber
T. L. Hawk
R. Hodges
W. R. Kampe
B. V. Koen
N. Leal-Cantu
H. S. Marcus
P. F. Meyfarth
G. Piotrowski
T. Pitidis-Poutous
K. T. Pogran
H. A. Radzikowski, II
C. C. Tillman, Jr.
C. Weissgerber
P. A. Wieselmann

Guest

P. Anesse - U. S. Navy

Project Intrex - C. F. J. Overhage, Charles H. Stevens, Joseph J. Beard, Peter Kugel, Robert L. Kusik, Julie B. Lovins, Richard S. Marcus, Richard Domercq, William Kampe, Tony Lin, Stephen Stuntz

Project Intrex (Information Transfer Experiments), a program directed toward the functional design of new library services, is continuing its activity under the direction of Professor C. F. J. Overhage. The project has been established with the twofold objective of finding long-term solutions for the operational problems of large libraries and of developing competence in the emerging field of information transfer.

The Electronic Systems Laboratory, under its Director, Professor J. Francis Reintjes, is currently pursuing two areas of experimentation in support of Project Intrex — the augmented catalog experiment and the remote-text-access experiment.

A. THE AUGMENTED CATALOG

The research program to create and test an augmented, machine-stored, remotely accessible catalog of library holdings is sponsored by the National Science Foundation. The purpose of this experiment is to learn from users of the catalog their preferences for types and formats of catalog information and to discover the catalog elements that are required by a reader in order to reach a decision to seek or not to seek the cataloged document in the library.

Preparation of the catalog is going forward in three groups within ESL — the Catalog Input Group, the Software Group, and the Hardware Group.

The Catalog Input Group in the past year has identified the literature of interest to them, in the area of materials science, has developed the catalog entry elements and entry formats, and has to date cataloged 1,400 of the initial 10,000 items planned for the experiment.

The Software Group has designed the storage and retrieval system and has implemented many of the basic computer programs that will be needed by the Intrex staff for "in-house" trials of the augmented catalog.

The Hardware Group has developed specification and design criteria for the catalog console. The Group is presently purchasing components for the console.

B. TEXT ACCESS

In the other experimental area, the Text Access Group is attacking the problem of providing the library user with guaranteed access to the material in the collection. This effort is being funded by a grant from the Council on Library Resources. The emphasis in this work, as in the augmented catalog program, is to learn from the user how the library

and the information system can best meet his needs. Access to both hard and soft copy is being studied: we plan to store full text on microfiche, and the group has already demonstrated transmission of both black and white and color images from microfiche to a remote screen.

Other experiments in the Intrex program await the larger funding that will make such efforts possible. Meanwhile, under partial sponsorship by the Office of Education, the Engineering Library is now undergoing a thorough renovation to make it suitable for the dual roles of a traditional high-quality library and an experimental facility for Project Intrex.

The MAP System - Roy Kaplow, John W. Brackett, Barbara F. Boudreau, and Henry Radzikowski

MAP is an experimental, but operationally interactive, on-line system for numerical analysis. While serving as a vehicle for the investigation of the design and implementation of "conversational" interaction in numerical problem-solving, it has simultaneously proved to be of widespread use in teaching and research.

Because of the imminent replacement of CTSS by Multics as the host time-sharing system at Project MAC, a major portion of our effort during the past months has been oriented towards the design of MAP II, which is to be implemented within Multics. In this connection, a number of alterations have been made in the internal organization of the present system to test their feasibility. These studies, as an important side benefit, have led to improvements by factors of 20 to 50% in the efficiency of various parts of the present system, mainly by reducing the number of core swaps and disk accesses.

An extensive in-depth study of four high-level systems for on-line mathematical assistance (Culler-Fried, Lincoln Reckoner, Amtran, and MAP) has been prepared (jointly with Adrian Ruyle of Harvard), in an attempt to provide a description of the current status of such systems and guidelines for the design of future systems.

The design of a modified version of MAP for CTSS has been established and partially programmed to study some new concepts in greater detail. Although limited by the present time-sharing system, the design includes provision for many of the features ultimately desired, such as two- and three-dimensional and complex functions, a more general but more efficient data-handling structure, and a more sophisticated, operator-based interpreter.

Although many of the inefficiencies in the present MAP would disappear in the Multics system, it is clear that this new time-sharing system will itself determine certain optimum design criteria. We have,

therefore, been considering the novel features of Multics in detail, to determine such subsystem guidelines, and Professor Kaplow and Dr. Brackett were among the instructors (together with Professors Hansen and Greenberger) in a Multics seminar given for credit to graduate students during the Spring term, 1967.

A second major effort during the past year has been the implementation of the MAP graphical-output requests on the ESL prototype low-cost display terminal (ARDS-I). All such requests, which provide for the display and comparison of one-dimensional functions, are now operational at the terminal in the Materials Center Computation Facility. The high-level command language allows the apparatus to be used by inexperienced users. For example, the command

compare storage a(x) b(x) log-log 2.7 10.3

would plot the function b(x) vs a(x) on automatically scaled log-log graph paper over the range $2.7 \leq x \leq 10.3$. The various required and optional parameters need not be memorized, since a command, such as

plot storage

will invoke a "conversational" interaction to obtain the necessary information from the user.

Thirdly, the mathematical capabilities of the present MAP System have been augmented, most notably by the addition of a matrix operation package. The package includes the operations invert, add, subtract, multiply, divide, determinant, set row, set column, input matrix, print matrix, transpose, adjoint, eigenvalues and eigenvectors. Scalars and vectors may be intermixed with matrices in appropriate commands, e.g.,

multiply A B C

multiply A b(x) C

multiply A b C

mean the multiplication of the matrix A by, respectively, the matrix B, the vector b(i), and the scalar b; the result in each case is the matrix C. As always within MAP, partial or erroneous commands will invoke computer interaction.

One of the most serious fundamental problems associated with interactive systems for mathematical analysis has been the typing required to input tables of numerical data, particularly when the values to be input are created by the user during the interaction. We have therefore constructed a large (~18" x 28") but relatively inexpensive graphical-input board which allows hand-drawn or curve-traced data to be input directly through the typewriter terminal; the user inputs a curve rather than tabular values. At present, the curves must be drawn very slowly,

since the operational speed is still limited by the typewriter; this deficiency can be eliminated, however, and in any case does not seriously interfere with our study of the interactive process. Development is also underway, utilizing the input board, of software for drawn symbol recognition to allow input of commands as well as data.

The TREC System - Kent F. Hansen and Billy V. Koen

The TREC system has been developed to permit user access to complicated nuclear codes via console communication. Input data is collected through a question-answer dialogue which relieves the user of any need to know the details of the code operation, data formats, etc. Previous progress reports have described the structure of the system and the types of codes available.

The system is in routine use at the Computation Center and has been used for homework, thesis research, and demonstration purposes. The major addition to TREC in the past year has been a monitor to control access, protect the system, and perform routine time accounting. The system resides on the disk under one user account number, while access is achieved through links from another user number. There are provisions in the monitor for the system administrator to add, delete, or modify users. The monitor program has been made available to other projects for use with their own multi-user systems.

The present method of linking to the system has a serious defect in that users of the dummy account number are not limited just to using TREC. At present there is no mechanism in CTSS to restrict users to only certain commands. The temporary solution adopted is to limit the dummy user memory space to only one track on the disk. Hopefully, the access-control mechanism in Multics will help correct this problem for the next generation of TREC.

Chempos: A Chemical Engineering Language - Daniel L. Flamm and Kent F. Hansen

Chempos will be a chemical-engineering problem-oriented language. Initially it will be process-oriented, but later implementation of purely mathematical facilities is contemplated.

The user of Chempos will specify, in terms of a flow diagram, a sequence of operations that comprise a process. This process could be a chemical plant synthesis, e.g., a particular ammonia plant, or a sequence of mathematical calculations (for, say, a heat-transfer problem). The unit operations of Chempos will exist as already-written skeleton routines into the AED language to be tailored, by the Chempos Processor, to the user's needs. The Processor may insert or delete lines, data and

linkages into the "skeletons" and will assemble the resulting programs into a process with Boolean matrix techniques. The user will also be able to insert operations of his own, providing they are written in an appropriate subset and format of the AED language. The output of the Chempos Processor, an AED program, will then be submitted to the AED compiler and be translated into machine code.

"Descendants" of the first Chempos will be made in such a fashion that the process flow diagram can be modified during execution without repeating the translation process.

During the past year a definite plan for implementation of Chempos facilities has been formulated. It was decided to first develop a suitable operation in the form of a skeleton program that can be cascaded by itself. Distillation was chosen, because a Chempos with distillation alone would be of practical importance and serve to underscore such a system's potentialities. Distillation programs are notoriously time-consuming; an obstacle that must be overcome if an iterative network of distillations is to be solved. An algorithm, for what is hoped will be a fast-running distillation, has been developed and the program is nearly completed.

Following completion of the distillation program, an algorithm designed to dissect flow diagrams will be implemented and used with a third part of the preliminary plans, an iteration algorithm to solve the resulting simplified network of cascaded distillations.

Subsequent work will add more operations, evolve more sophisticated command language. and develop graphical input-output facilities.

Optimization Methods Applied to Ship Design - Philip Mandel and Reuven Leopold

Drawing on techniques developed in other fields of engineering, an optimization method useful in the preliminary design of ships was developed. A convergent random-search technique, incorporating an exponential transformation, was examined in detail and chosen to be used in association with a weighted multiple-parameter optimization criterion. Rather intensive application of the method was made to a cargo-ship design model. The versatility of the chosen optimization method was demonstrated by its application to a decision problem under uncertainty, using a model of the tanker design process. The features of the chosen optimization method are that it 1) arrives at a solution in less time than current methods, 2) is more versatile than any other method available, and 3) performs the optimization more correctly. Future work will apply this method to the design of specialized ships, using more sophisticated economic criteria.

Support for this research was provided by the U.S. Navy Bureau of Ships, Contract No. Nobs 90100, as well as by Project MAC.

The Computerization of Series 60 Resistance and Propulsion Experiments - Theodore Pitidis-Poutous

The object of this investigation was to computerize the Resistance and Propulsion estimating methods contained in Report 1712, David Taylor Model Basin. This was accomplished by fitting the original experimental data in 5-dimensional space, using Chebyshev polynomials. A program which will evaluate the equations obtained in this way and which will give estimates of EHP and SHP for a ship, based on Series 60, has been written and tested. One percent average deviation from the Series 60 report methods should be expected.

Toward General Stress-Strain Analysis - Frank A. McClintock, Robert Hodges, and Carl Weissgerber

The analysis of metalworking operations, the calculations of the strength of designed parts, and the prediction of fracture would all greatly benefit from a facility for stress analysis of arbitrary shapes. Typical situations are so complex that solutions can only be obtained by numerical methods, and at present there exist very few solutions.

The technique under investigation is to progressively modify assumed displacement fields in the light of the resulting unbalance of forces. Early efforts to obtain a solution involving third-order, non-linear equations by iterative techniques have proven too difficult as a first step. A mathematically simpler problem (cylindrical inclusion in transverse shear) allowing primary concentration on the iterative modification procedure has been almost completed using EPS SAVED, a program developed at MAC by Coyt Tillman. At present, programmer/console interaction must still be included in each iteration. Reliable criteria for modifying the extent of the elastic zone from consideration of boundary stress unbalances are being evolved, however. A preliminary outline has been made of an instruction manual which would allow the use of EPS SAVED as a general problem-solving tool without requiring user programming skill.

Equilibrium Problem Solver (EPS) - Coyt C. Tillman

In previous progress reports, a system for solving elliptic boundary-value problems has been described. This system exploits the on-line keyboard and graphical-display consoles available through CTSS to provide a general, flexible tool for engineering design and research.

The past year was spent on extensive testing of EPS and on the compilation of a series of case studies for a thesis now being prepared. Test results have in general been most favorable, showing EPS to be applicable to a broad spectrum of practical problems. Case studies include examples from the fields of heat transfer, fluid mechanics, soil

mechanics, and stress analysis. Examples have involved the treatment of several differential systems, ranging in complexity from a single Laplace equation with Direchlet boundary conditions to the Von Karman plate equations (a non-linear system of four second-order partial differential equations) with mixed boundary conditions. One of the more challenging examples (see Meyfarth, this section), has involved treatment of the Navier-Stokes equations for steady-state viscous fluid flow.

Solution of Viscous Fluid Flow Problems - Philip F. Meyfarth

Work on the solution of potential flow problems was reported in the last progress report. As an extension of this effort, work was done to explore the applicability of EPS (see Tillman, this section) to the solution of viscous fluid flow problems. The steady two-dimensional Navier-Stokes equations were applied to boundary-layer flow on a flat plate and to branching flow in a T-shaped region.

Historically, the Navier-Stokes equations have proved quite difficult to solve: they are quite non-linear, and can easily produce numerical instability. It was necessary therefore, to develop a technique by which the desired solutions could be obtained, while avoiding instability on the one hand and excessive computational time on the other. The graphical output capability of EPS permitted rapid visualization of the developing solution, in particular the onset of instability. This, coupled with the interactive nature of EPS, permitted reasonable control of the solution process.

For low-Reynold's-number flows (i.e., creeping flows) reasonable agreement with known solutions was obtained with this technique. As expected, however, at higher Reynold's numbers (i.e., in the direction of increased turbulence) considerable difficulty was experienced due to numerical instability.

Two-Dimensional Stress Analysis - Paul A. Wieselmann

Plane stress analysis by the complex-variable method (or method of Muskhelishvili) rests entirely on using a function to map conformally the region of interest into a canonical domain where the problem is in fact solved. When viewed as providing a base for numerically solving fairly general problems by a single, easy-to-use system, there is an additional need for the mapping function to be in a form consistent with the way the subsequent calculation is to be handled. The complex variable method admits of some lattitude in the choice of the kinds of calculations and the numerical implementation broadens this even further. Therefore, it has been the intent of this work, as evidenced in the past several progress reports, to explore the advantages of a few tractable methods of obtaining mapping functions and a few tractable methods of solving the

complex variable formulation of the stress analysis problem. It is a pleasure at last to report the fruition of these studies in a coherent system of programs to solve the first and second boundary-value problems of plane elastostatics in simply connected finite or infinite regions. The system also possesses the special advantage of being able to treat boundary cracks in a precise manner as part of its normal operation.

A synopsis of the system is as follows. From a point description of the boundary, the Schwarz-Christoffel integral for the exact mapping of a polygonal boundary is obtained. A suitably accurate Fourier expansion of this integral is then made. Special terms may be introduced into the series to give it precise behavior at specified points, e.g., in studying stress concentrations at corners or at crack tips. The boundary conditions are specified by the user as tractions along segments and discrete loads or as displacements along segments. With the aid of the mapping function, the problem is solved on the unit circle, as a canonical domain, and the system will open itself to interrogation about the stresses or displacements. At all times the system will be interruptable for interrogation and assistance by the user, or more importantly, for direction, e.g., when load or boundary perturbation studies are made.

The overall operation of the system and all the numerical schemes have already been planned. The final programs are about fifty percent complete. A doctoral thesis in the mechanical engineering department and a MAC report and user's guide will be forthcoming in the next reporting period.

ENPORT: A Multiport System Simulator - Ronald C. Rosenberg, Roger Banks, and Nestor Leal-Cantu

ENPORT is a digital computer program that accepts a bond graph description of a dynamic system and produces its time response. As the number of available algorithms for processing bond graphs grows, the capabilities of ENPORT can be augmented.

ENPORT-1, reported in last year's progress report, accepts linear, active and passive, lumped-parameter models. A senior in mechanical engineering at Tufts University, George Mayfarth, used ENPORT-1 to help explain the observed behavior in a hydraulics experiment. The project won second prize in the Region 1 ASME Student Paper Competition this past spring.

ENPORT-2, which is currently being coded and debugged, has been reorganized to use the graphical transformation procedure described in Modeling and Analysis of Multiport Systems, this section. Thus a rather broad class of non-linear, active and passive, mixed-energy-type systems can be handled. Peter Weindahl contributed to this task. In addition, a file structure to permit both temporary and permanent storage and

retrieval of arbitrary multiports is being implemented. This work is being carried out as part of a project to develop computable bond graph models for a class of engineering components.

Once the state-space equations are formulated, computation of trajectories is accomplished by matrix exponential techniques if the system is linear. A procedure is being developed to ensure that the calculated state at every stage is accurate to a specified tolerance. Also in progress is the testing and implementation of a procedure given by Koepcke for the computation of state time response in a system governed by linear differential-difference equations. Such equations arise when a physical system model has both lumped and distributed parameters.

As the average size of bond graphs described to ENPORT increases, due to improved simulation abilities and storage facilities, use of a graphical communication device becomes more important. Anticipating this requirement, Stephen Braunstein wrote a program for bond graph input-output on the ESL KLUDGE display, which includes processing of arbitrary multiports.

Applications of ENPORT-2 to problems in mechanical vibration and control, and to the study of fluid systems behavior, are planned as soon as the program is operating satisfactorily. Development of computable models for classes of engineering multiports will make ENPORT-2 useful for complex design studies, as well as providing an interesting storage form for component models.

Modeling and Analysis of Multiport Systems - Ronald C. Rosenberg

The multiport representation of engineering components in terms of their energy exchanges promises several advantages over conventional techniques. The bond graph notation, using powers and generalized energetic elements as building blocks, has been applied in representing engineering components and systems of several energy types — electrical, rotary and translatory mechanical, and hydraulic, to name the most common types. An example of a multiport model expressed as a bond graph is given in Figure 38. A simple pump model which assumes incompressible flow is shown. Such a graph reveals a high degree of structuring of the model, even if many elements are non-linear.

Work has continued on the theory of bond graphs, and a systematic graphical method has been developed for generating the governing equations of a broad class of non-linear systems. The procedure, illustrated by an example in Figure 39, consists of first substituting for certain bond-graph elements their mathematical equivalents in terms of a primitive set. The graph of primitive elements is then cast into a standard form, from which the state-space equations may be written by inspection. One interesting

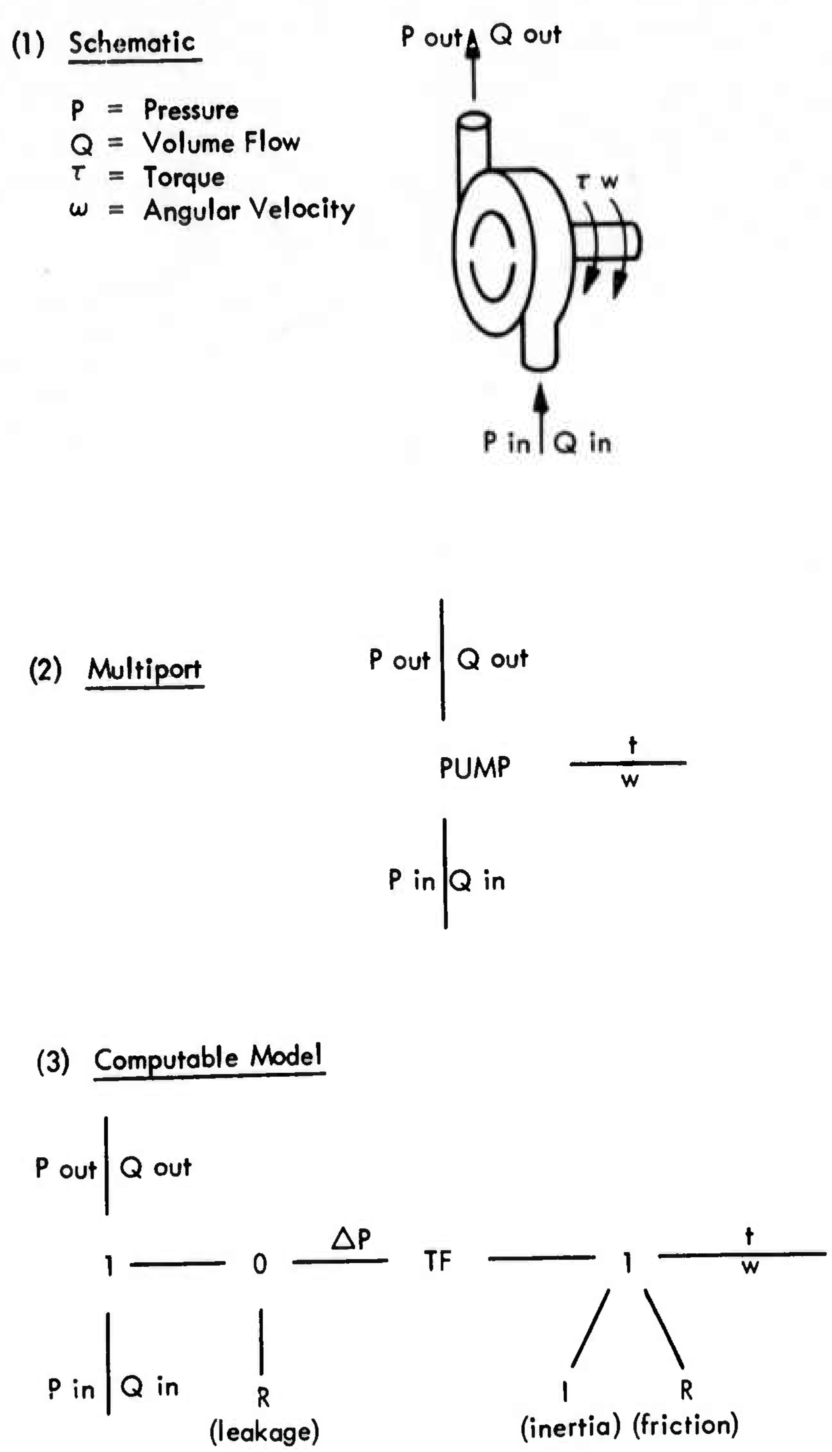


Figure 38. Bond Graph Model for a Pump

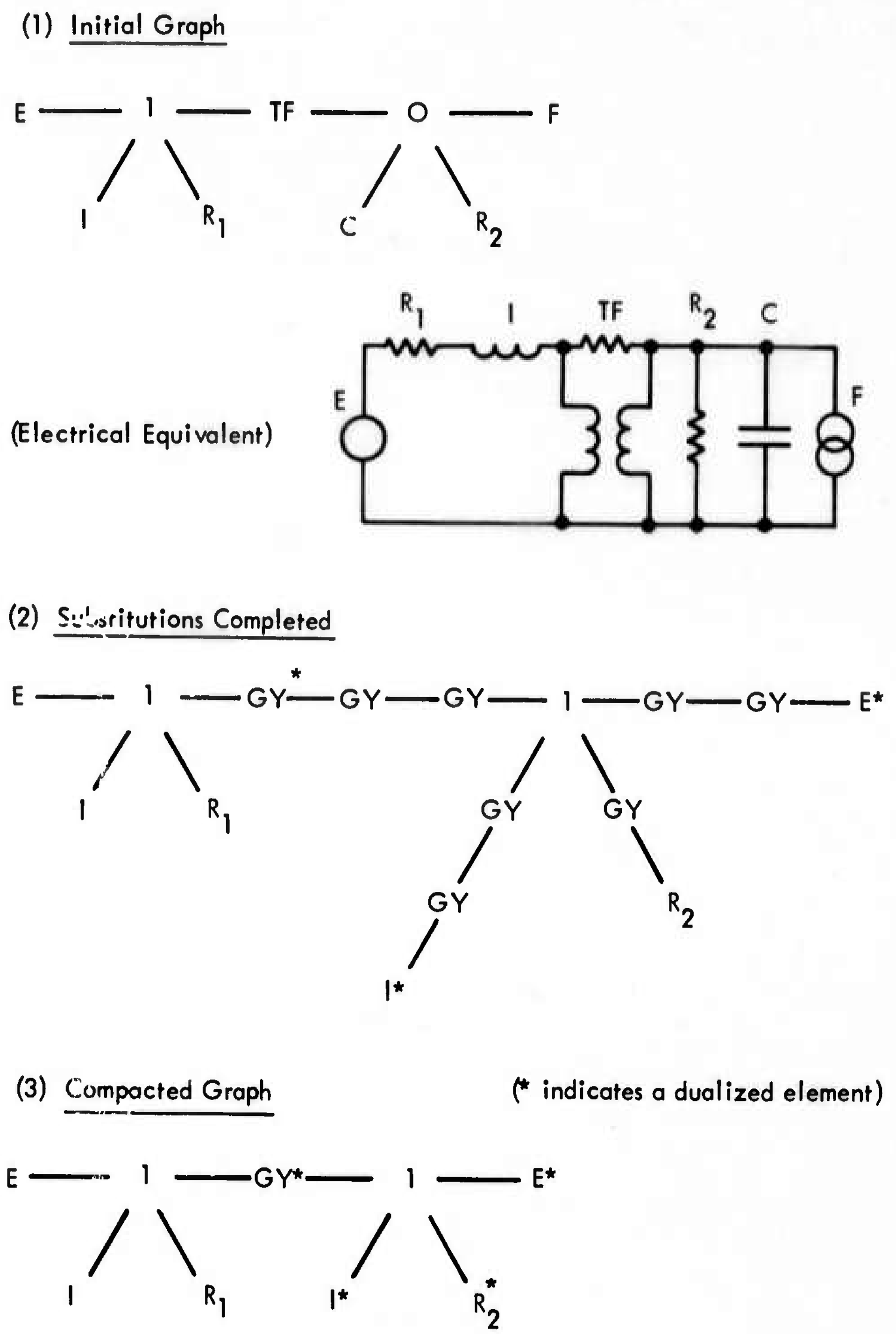


Figure 39. An Example of the Transformation of a Bond Graph to Standard Form

aspect of this procedure is the way in which the representation of fields occurs. Insight into system structure gained by using the method has proven useful in generating bond graphs from equations; for example, see Figure 40. A potentially fruitful area of investigation is the automated synthesis of bond graph models from experimental data. Some preliminary work has been done on this topic (see Free, this section).

Identification of Non-linear, Lumped-Parameter Models for Dynamic Systems - Joseph C. Free

This research has been aimed at developing algorithms which operate on the transient-response record of a dynamic system to determine the non-linear, lumped-parameter characteristics of a given structural model. The models considered are those which may be represented by the first-order matrix equation

$$C(\underline{X})\overset{\circ}{\underline{X}} + G(\underline{X})\underline{X} = B(\underline{X})\underline{F}$$

where $\underline{X}$ is a vector of state variables; $\underline{F}$ is a vector of inputs or forcing functions; and $C(\underline{X})$, $G(\underline{X})$ and $B(\underline{X})$ are square matrices whose elements are non-linear functions of the state.

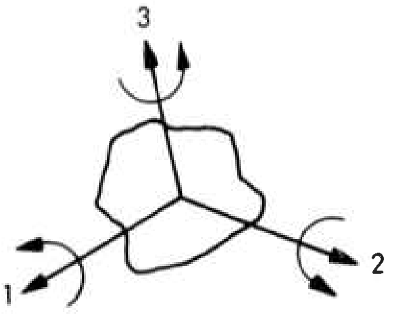
The nonlinearities are restricted to single-valued continuous functions of a single known argument. That argument may be a linear or non-linear single-valued function of the state variables. The nonlinearities are modelled by piecewise linear, piecewise cubic, and arbitrary polynomial functions in this research, although the algorithms may apply to other useful functions, such as Fourier sine or cosine series.

The method requires a history of all the state variables. If they are not explicitly available, then filtering techniques may sometimes be used to estimate them, although the generality of the modelling equations becomes greatly restricted.

The techniques are being verified in a computing "laboratory" where they are applied to simulated non-linear prototype systems. Detailed results will be given in a Ph.D. thesis to be presented in the Mechanical Engineering Department, September 1967.

(1) The Problem

Rigid body rotation in principal axis coordinates



(2) Euler's Equations

$$I_i \dot{\omega}_i = (I_j - I_k)\omega_j \omega_k + \tau$$

where $i, j, k = 1, 2, 3$ cyclically

(3) The Bond Graph in Standard Form

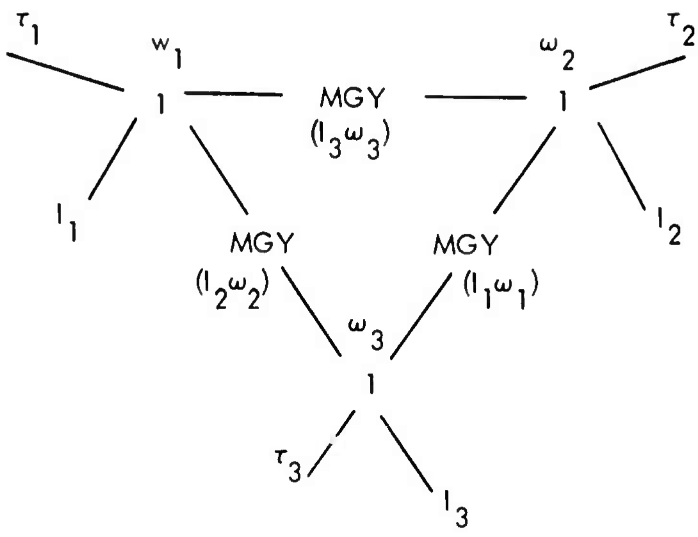


Figure 40. A Bond Graph Representing Rigid Body Rotation about Principal Axes

BLANK PAGE

SCHOOL OF HUMANITIES AND SOCIAL SCIENCES

Programming for the ECONOMETRICKS Project

The Design of an On-Line, Large-File, Social Data System

Computer Models of Soeial Processes: The Case of Migration

Social Distance: A Network Approaeh

ADMINS - An Administrative System for the Management of Social Data

Models of Turkish Attitudes

Analysis of the Turkish Peasant Survey

CRISISCOM Simulation

Conflict and Consensus in Venezuela

VENELITE Biodata Analysis

Academic Staff

J. M. Beshers
F. Bonilla
J. DaSilva
C. Domingos
E. Kuh
D. Lerner
J. S. Michelena
I. d Pool
L. L. Roos, Jr.
J. Waelbroeck

Non-Academic Research Staff

M. Eisner
S. D. McIntosh
J. Rizika

Research Assistants and other Students

P. Berman
P. Bos
T. Carroll
F. H. Ciaramaglia
P. B. Clark
R. Crossley
J. L. Fields
M. Fineman
J. E. Gips
A. J. Gottlieb
D. M. Griffel
K. Grossen
C. P. Hensler
D. A. Hendrick
A. R. Kessler
J. C. Klensin
J. F. Kramer
D. W. Levine
T. E. McGettigan
J. T. McGettigan
R. McKelvey
P. V. Mockapetris
A. Moulton
P. M. Raup
G. S. Ruderman
H. L. Selesnick

Programming for the ECONOMETRICKS Project - Edwin Kuh

TROLL (O), which is the aeronym for the programming of the ECONOMETRICKS Project, has been designed to provide a reasonable working environment for econometric research on the CTSS time-sharing system at M.I.T. TROLL (O) is the forerunner for a complete subsystem to be implemented on the Multics system and also on a suitable system for the IBM 360.

TROLL (O) will be divided into a series of logical phases, each of which will be further subdivided into a series of modules designed to perform a specific system activity. At present, the system has been divided into four standard phases – a data processor, an estimator, a simulator, and a specialized phase for the library facility. Figure 41 portrays the functional interaction between these phases and Figure 42 portrays the path of data transfer between different phases. Each of the phases carries out a specific function related to econometric research:

1. The data processor's function will be to process econometric data, editing and checking until the data is in useable form. This phase will also be primarily responsible for the retrieval of data from the library and its proper display to the user, including a table-building facility, a plotter, and a display scope package.

2. The estimator is designed to estimate coefficients for econometric models. It will include standard methods for estimation, as well as several new estimating procedures and a set of statistics for verifying results.

3. The simulator is expressly designed to perform time-dependent simulation of different equation models. It will allow a great deal of flexibility in setting up models, editing, performing simulation experiments, and monitoring the results of the simulation. The simulator is a vital, but currently weak, link in econometric research and, therefore, considerable effort is being concentrated on it. The simulator will later be redesigned to operate in two ways: the first to give model solutions, the second as a Monte Carlo device to evaluate the small-sample properties of statistical estimation procedures.

4. By its nature, the library phase differs from the other parts of the system. It does not perform an explicit function for the economist, but instead functions simply as a thoroughly documented information source. It will allow the user to search through a hierarchical index for information and then deliver copies of the information (in tabular, graphic, or scope form) to the user. The library will contain two sub-sections: a data

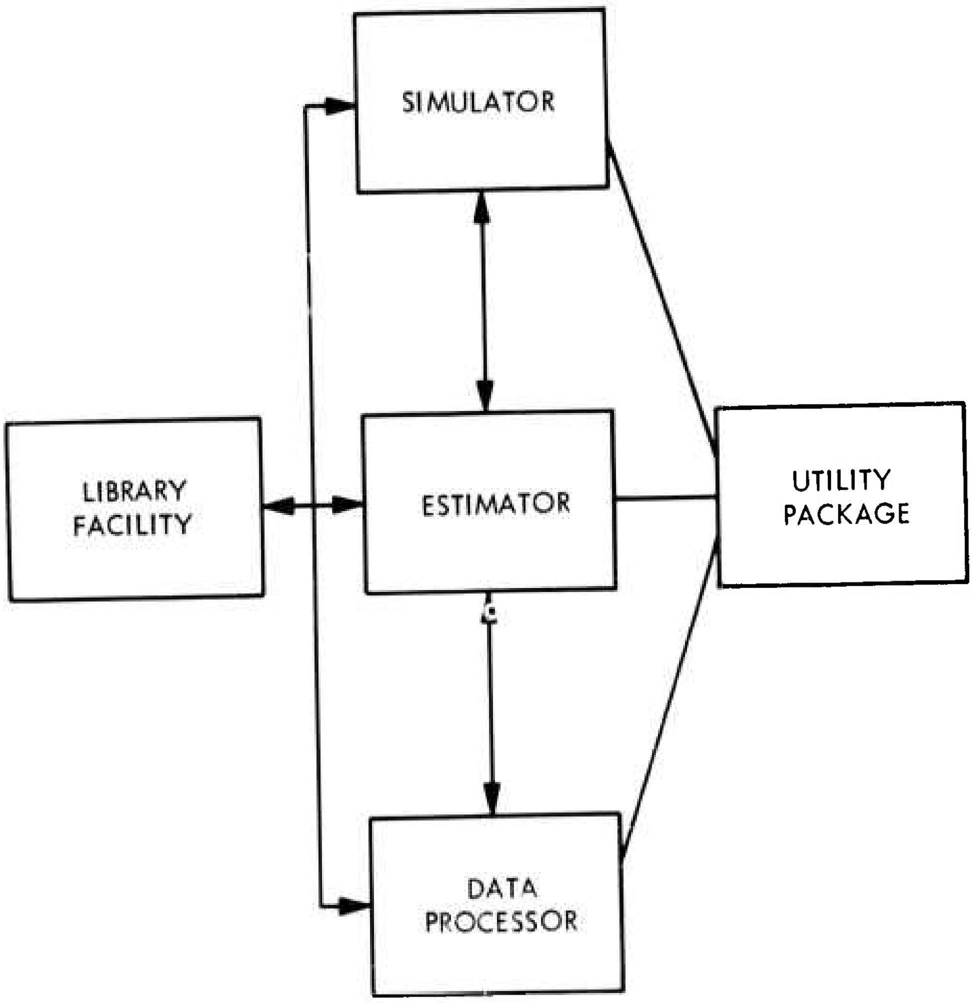


Figure 41. TROLL (O) Processor Interaction

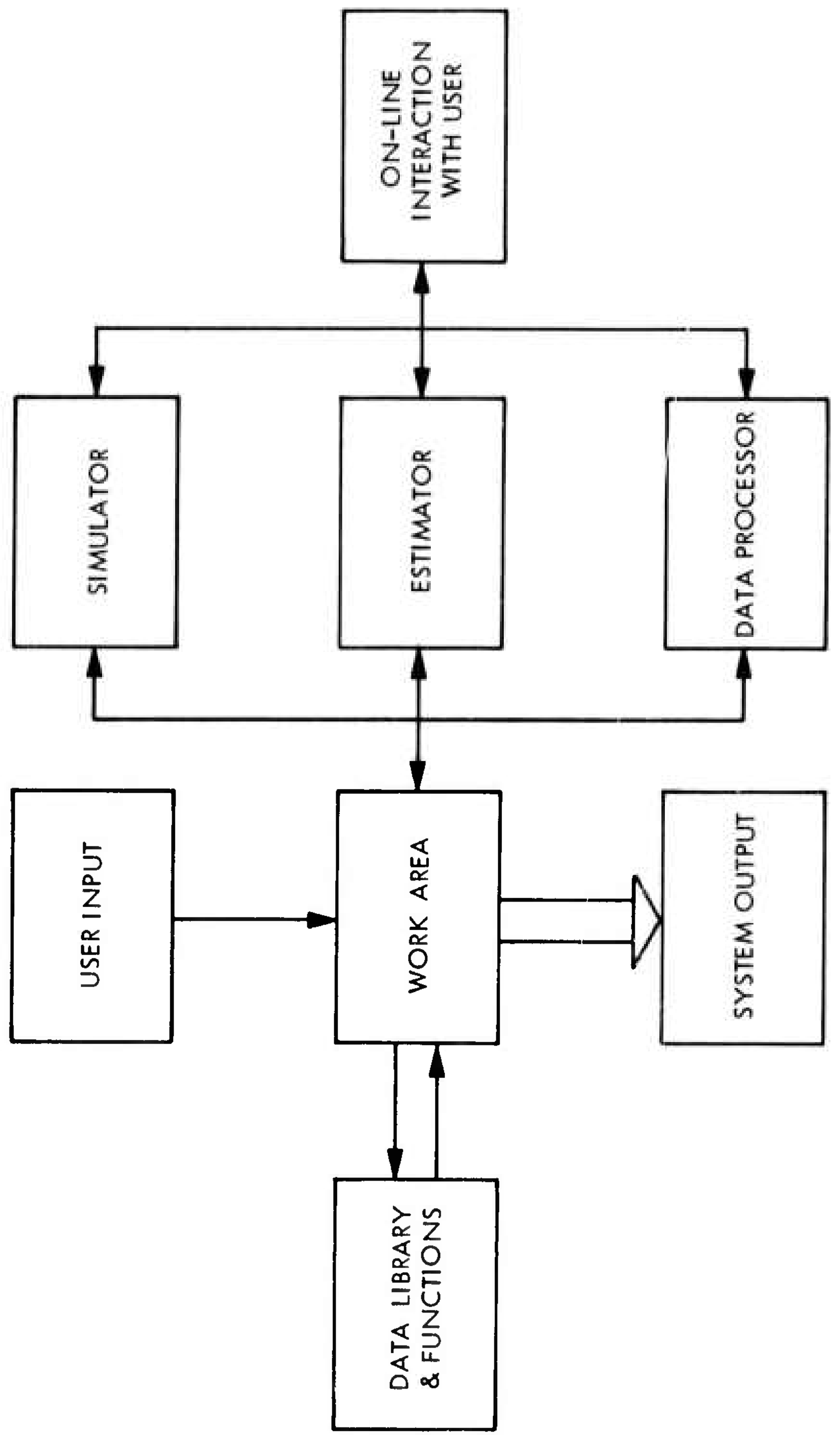


Figure 42. TROLL (O) Data Movement

area and a function area. The data area will contain, in general, time series and survey information. The function area will contain copies of standardized functions, which an economist may use as tools during his experimentation. It can be seen in Figure 42 that all information from the library goes to a working area, which in turn can be accessed by the user and the other phases of the system. The ability to enter things into the library will be restricted so that only complete, debugged, and useful information can be entered. Only a restricted number of users of the system will have permission to enter information into the library, as a necessary condition for maintaining quality standards for the data.

In turn, a phase will be divided into three types of modules:

a) A processor will do the actual work for the system. There may be several of these contained in each phase;

b) An editing mode will provide for the entering and editing of both symbolic and numeric data;

c) A control mode will be responsible for the primary interface between the phase and the user, and will also provide the only communication between its phase and other phases of the system;

d) A diagnostic mode will contain, besides the standard phases described above, a large utility package. This package will serve two functions. First, it will provide a pool of often-used routines to avoid redundant coding. The pool would contain such things as the basic I/O, and the string manipulation processor. Second, and far more important, the utility package will contain the routines that make up the sub-structure for the system.

These are the machine-oriented routines that will create the basic environment in which TROLL (O) will function. Specifically, the utility package will provide the paging system which will simulate a Multics-like paging scheme for CTSS and loader/unloader facility for the rapid handling of phases in the system.

This project began last fall with Mark Eisner as chief system programmer. Design work was undertaken until January when a part-time staff of six M.I.T. undergraduates was assembled. The system will be programmed in AED throughout. At this time, the programming staff has acquired basic familiarity with AED, and several important components including the string manipulator, model input language, and graphical display have been completed and debugged.

A large-scale effort this summer by eight full-time programmers and two econometricians is expected to culminate in a running system available sometime during the Fall of 1967. The library design is being worked on collaboratively with Aaron Fleisher and Wren McMain, City Planning Department, and in consultation with Professor Richard Ruggles of Yale. This project is supported in part by NSF Contract GS-1376.

The Design of an On-Line, Large-File, Social Data System* - James M. Beshers and Stuart D. McIntosh

With a computer-based, on-line, large-file social data system it is possible to bring theory construction and data manipulation into a single coherent domain of activity - concept formation, system modelling, and "verification" may be served by one computer system.

In such systems, one must provide for natural, even idiosyncratic, languages of discourse. Two principles emerge in the design of the discourse: 1) the design must allow for learning - for reconceptualization of a theory; and 2) the design must provide criteria and techniques for eliminating arbitrary systems. Thus one must be able to compare alternative systems and to prefer them for their parsimony (a property of the theory), or for their relative empirical adequacy.

Such systems must allow simultaneous communication between different sets of data and different conceptual systems. This means that data formats must be parameterized and that conceptual systems must be parameterized. The "indexing" problem thus rests on the definition of transformations of social concepts; the Boolean, or set, operators implement such transformations in the context of explicit social system models.

It is clear that the heart of such a system lies in the primitives that are available to the user. These primitives must underly the archive, or system, for the indexing and manipulation of the data, and they must underly the system modelling at the user interface. The archive system is being designed by Stuart McIntosh; I am working on the user interface.

From the user perspective there are three activities that require a parallel development. These are the assignment of labels to categories, the design of the conceptual system underlying the categories and their

---

*The research described in this report is supported in part by NSF Grant GS-727, Ithiel de Sola Pool principal investigator, and by NSF Grant GS-1043, James M. Beshers principal investigator. The facilities of Project MAC, of the M.I.T. Computation Center, and of the Harvard Computation Center have been used in this research.

relations, and the design of causal representation in the dynamic modelling of social systems. All three of these activities must be contingent on the state of the data: appropriate feedback must enable the user to modify the systems in light of data.

The individual user needs to compare alternative systems of his own creation. Further, one must compare the systems of several users. This means that idiosyncratic labels may be assigned at the convenience of a given user, but that the primitives of the system must define transformations that enable us to determine how similar two or more systems are. Similarity of systems may be defined by output characteristics, i.e., the deduced time series from a dynamic system model, or it may be defined in terms of structural properties of the systems. In the former case only those systems can be compared that adhere to some common output, or for which there is an explicit transformation algorithm that defines comparable output. In the latter case there must be an explicit notation to describe structure.

The relation between the conceptual system underlying the categories and the dynamic modelling of social systems is subtle. On one hand it is clear that the model maker must go through the category system in order to communicate with the data. Thus there must be transformation algorithms to accomplish this task — in effect there must be a way in which the user can learn to communicate through the category system and, in doing so, change the category system. On the other hand it is clear that subtle differences of purpose can have profound implications for dynamic model construction. Strictly speaking, we wish to make three distinct applications of one system of dynamic simulation models: 1) analysis and interpretation of historical trends, 2) the projection or forecasting of future trends, and 3) the simulation of the effects of alternative policies. For each application there must be communication with data, but the perspective and purposes of the user differ greatly in each case.

My current application is building computer models of internal migration (as a non-stationary stochastic process) that may be associated with a metropolitan information system (or data bank). In this application it is evident that the elements are (persons) X (addresses). One must update these elements from migration information. Some concepts may be defined by aggregating (taking the union) over such elements; we can define other concepts as measures associated with these concepts. Thus persons may belong to the same household, while addresses may be combined into areas, and households may have total incomes, or areas may have average incomes.

From a subject-matter viewpoint we find that certain types of official data systems play a central role in data organization. In the U.S., the Bureau of the Census plays such a role — census concepts and census

data must be appropriately integrated into the data system. Other data sources (surveys or record systems) contain some concepts that were intended to be similar to census concepts; these become critical links in the information system.

The implementation of such a system involves on one hand the notation and conventions of graphs and of matrix algebra, and on the other hand list-processing capacities of a computer. (See Beshers, Appendix C.)

Computer Models of Social Processes: The Case of Migration – James M. Beshers

Papers entitled "computer models of such and such ..." have a way of being rather disappointing. At worst people will hold up a number (or perhaps some letters spelling the word 'births') that fell out of a computer and say "This is the number of births." They will have had some conversations with a programmer, perhaps paid him to work on a project, and one day was given some computer output and told, "Your program runs." Some euphoric text about the world of tomorrow is then wrapped around this computer output and a paper is presented at a 'scientific' meeting. (See Beshers, Appendix C.)

Critical issues of model building apparently are not recognized, certainly never discussed. How should the output be interpreted? How many different kinds of output are available, and how are they to be compared? What are the input options? What is the mathematical structure underlying the model? What special computer techniques were required to implement the model? Above all, what has experience shown in the use of the model (such as sensitivity tests of the parameters) — if one run of a model tells all, then there is not much to tell.

This work is described in the light of these questions. First, the overall system for building models is described and then illustrated with a set of models built for research on migration — the Demographic Model. In all that follows I have had primary design responsibility and have done much of the programming myself; I am also the current programmer on the project. Many people have given me essential help, especially Donald Akers and John Bidwell on the birth component, and George Leyland on the present structure of the main program as well as on the migration component.

The fundamental issue in current computer programming systems is user control that facilitates user interpretation of computer output — in particular the relation is seen as a continuing conversation, a dialogue between user and computer system. This means that the essential features of the computer system must be interpretable in social science concepts — the structural features of the model, the range of alternative

forms of the model and the alternative outputs of the model.* In the context of the on-line, large-file computing systems at M.I.T. this user-controlled communication takes on many new forms.

My system for building models is oriented to three kinds of application: 1) projections, or forecasting, of future trends; 2) analysis and interpretation of historical trends; and 3) the estimation of the effects of alternative policies upon social systems. The same mathematical structure underlies all three of these applications; the differences lie in one's tactics in using the models, especially in the extent to which alternative sub-models reflect hypothetical or empirical considerations.

The program structure allows for three levels of generality in building alternative models. At the simplest level, alternatives can be defined through conventional input of numerical values (under a READ statement). At the next level, certain parameters can be specified, perhaps by control cards that precede the regular data input. At the third level, alternative subroutines, pieces of program, can be defined and called by a main program. All three levels are commonly used, even by inexperienced programmers. The central issue is, how do we design these levels so that the social scientist user retains control at all times?

User control can best be understood in the light of particular examples. The Demographic Model has been developed to aid migration research. Here we shall describe MIGM4. This version of the main program includes a birth component, a death component, a migration by social mobility component, and two feedback subroutines that allow the probabilities of migration and of social mobility to be updated — one feedback subroutine takes account of past history, the other takes account of present overflows. All components are subroutines — thus there is considerable flexibility in the assumptions for these aspects of model building. Further, being on-line at a console makes it possible to write new additions to the library of subroutines in about ten minutes and then try another run. Thus, the structure of the subroutines is kept simple.

To see the design more clearly we must look at the structure of the main program. The main program has been designed as a nonstationary Markov process that incorporates the cohort features of Whelpton.**

---

*See my Introduction to Computer Methods in the Analysis of Large Scale Social Systems, Joint Center for Urban Studies of M.I.T. and Harvard, Cambridge, Mass., 1965

**See Beshers and Reiter, 'Social Status and Social Change'', Behavioral Science, March 1963; and Beshers, "Birth Projections with Cohort Models", Demography, Vol. 2, for earlier descriptions of these features.

The major loops in the main program are simple and conventional. An outer loop increments on calendar time while an inner loop increments on age. Thus a distribution of the population by age and calendar date is a necessary feature of the program; further, this distribution is indexed by birth date so cohort-by-age distributions can be reconstructed. For every age-by-date cell there is a set of states of the system (six such states for developmental purposes). These states can be interpreted as areas, as social characteristics, or as a cross-classification of areas by social characteristics. Thus movement over the states of the system may be interpreted as migration, as social mobility, or as both. In the last case, migration and social mobility may be viewed as reciprocally influencing each other.

Let us assume that the states of the system represent areas. Then the probabilities of change represent migration. Now we can introduce the special features of the main program. Consider a distribution vector m(t) defined over these six areas, and a probability matrix (stochastic) P(t) that defines the probabilities of transitions over the set of states in the next time interval. The calculation of the new distribution vector is conventional, $m(t+1) = m(t)P(t)$, but the calculation of the new probabilities is not conventional — these are calculated in two separate subroutines. In both subroutines the issue is to represent lagged feedback relations, since the output of each subroutine is a new transition matrix.

The ASUB series of subroutines calculates the new probabilities as a function of the old probabilities and the two most recent distribution vectors — these are the probabilities and the distribution vectors associated with this age cohort. The calculation follows a rule that is specified in advance anc represented as mathematical equations. The choice of these rules becomes the critical issue in model building. Therefore much effort has been devoted to building a library of subroutines that can be readily understood by social scientists. One example is an attraction, or gravity model, subroutine.

The BSUB series of subroutines calculates the new probabilities as a function of constraints upon the distributions. For example, maximum numbers for each area can be specified; if these numbers are exceeded, then the probabilities of going to that area are decreased and the probabilities of leaving are increased. The maximum numbers can be constants or they can themselves be updated by a rule. The BSUB series considers distribution vectors summed over age categories, say, under twenty, between 20 and 65, and so on, whereas the ASUB series considers each age group separately.

Now let us turn to the two problems in migration that we are currently studying — migration among metropolitan areas (or State Economic

Areas) and migrating within metropolitan areas. In the former case we view migration as responsive to labor markets; in the latter case we view migration as responsive to housing markets as well.* In the former case we consider skill level and career as constraints; in the latter case race, ethnicity, and household characteristics are constraints. This means that the maximum numbers in the BSUB subroutines are calculated on the basis of job supply or of housing supply. Note that queuing notions are implicitly used here, and that a more complete definition of a market would represent the job supplies and housing supplies themselves as being in part dependent upon the demographic process (the joint effects of births, migration, social mobility, and deaths).

Note that the ASUB subroutines can be thought of as representing attraction or repulsion of various areas for various kinds of people. An explicit decision mechanism in the ASUB series characterizes persons in each area by the average of their preferences over all areas, including their own area. The preferences are expressed in terms of a utility number and a likelihood number — the product of these two is the expected return for the average person in each area if he locates in it. By letting the likelihood numbers be recalculated on the basis of experience, we include learning and differential distribution of information in the model.

At this time a new series of BSUB subroutines is being programmed that represents the constraints on job supply (for example) as constraints on likelihoods (for persons). (The next series of BSUB subroutines will represent demand for labor also in terms of the average utilities and likelihoods of employers in given labor markets.)

Now let us try to answer the questions posed in the second paragraph of this report. To gain specificity, let us discuss one application, namely population projections. This provides us with an interpretation of the output.

Alternative outputs are, in part, determined by the input options. In general the outputs are two kinds of summaries of the projected population. These are: 1) the population in each state of the system at each date (either total population or population by age); and 2) the population in each state of the system that belongs to the same age cohort (same birth date).

---

*See James M. Beshers, Population Processes in Social Systems, Free Press, New York, 1967, ch. 5

The input options are: 1) initial population distributions, age x state; 2) initial transition probabilities, age x state; 3) birth probabilities, age x state; 4) death probabilities, age x state; 5) rules for updating transition probabilities in light of cohort experience, age x state; 6) rules for updating transition probabilities in light of constraints on distributions (including initial constraints and rules for updating these constraints), age x state; 7) rules for updating birth probabilities, age x state; and 8) rules for updating death probabilities, age x state.

For intensive study of migration options, 5) and 6) are most important and have therefore received most attention in program design and development. These options specify the feedback equations by which the new transition probabilities are calculated in each time period. While the Markovian character of the whole system is preserved, it is clear that these lagged feedback equations are the crucial mathematical features of the model building system.

The computer techniques in the present model are not elaborate. The critical issues in model development were: 1) convenient storage and retrieval for the many distribution vectors and probabilities involved; and 2) design of the subroutines so that they communicate appropriately with the rest of the program and are easy to modify to obtain new subroutines (changing one, or only a few, cards). The solutions to these problems (at any given time) appear quite simple.

The simplicity of this program stems in part from the fact that we shall embed it in a large complex and novel system for data manipulation and concept formation. The preparation of data for input and the labelling of categories will be handled by the larger system. This system, ADMINS, was also developed at Project MAC. (I collaborated with Ithiel de Sola Pool who is principal investigator, and with Stuart McIntosh and David Griffel, who designed and programmed this system.) (See McIntosh and Griffel, this section.)

Experience with the Demographic Model shows that it is easy to represent migration as a process that: 1) proceeds rapidly to an equilibrium; 2) never stabilizes but instead fluctuates wildly; or 3) appears stable, abruptly changes and then appears stable, as in Duncan's "tipping point" concept for racial invasion and succession. Thus, it is clear that these possibilities are the important characteristics that differentiate classes of migration models, further focussing our attention on input options 5) and 6). Elaborate detail of social characteristics would not greatly improve the significance of our models unless it was shown that the feedback equations must be defined over this level of detail. Such detail could be incorporated into the present system, largely at the cost of much more computer time, but this temptation to "simulate the real world" should be resisted.

This last point can be seen if we consider the interpretation of computer output from a population projection perspective. Here we specify alternative assumptions as the nature of the migration process and then seek to determine the range of outcomes when present population characteristics are input. Our attention is to determine a reasonable range; in addition we can indicate that certain specified shifts in the transition probabilities that might result from certain feedback equations could lead to quite unexpected outcomes.

Now consider causal analysis as a perspective for interpreting computer output. Here we wish to compare the outputs of alternative models with various empirical time series in order to see which models, in some sense, come closest to the empirical data. I say "in some sense" since it is not likely that the conventional least-squares or goodness-of-fit criteria will be appropriate to select the "best" model, but rather that the response of models to input changes and parameter changes will more likely lead us to the "best". As suggested above, the manipulation of the empirical data will be handled in the ADMINS system. (We can do certain obvious comparisons within our present system.)

Now consider policy as a basis for interpreting output. Here it is extremely clear that response of the models to parameter changes is the interesting property of models. Alternative policies can be defined in terms of alternative values of parameters within one class of models. Alternative tactics for implementing policies can be represented by shifting models or parameters in different time sequences.

Throughout, I have not stressed the significant role of the Project MAC time-sharing equipment at M.I.T. in implementing these notions. The Demographic Model described here is programmed in MAD and can be adapted to either time-sharing or batch-processing computer installations. The ADMINS system for data handling, however, cannot be implemented save on time-sharing systems with large storage facilities. Further, the significance of the Demographic Model is far greater when used in time-sharing.

Empirical work in the context of these computer systems is moving forward. Before any of the systems described here worked, George Leyland and John Bidwell implemented some of this thinking in their M.C.P. theses. More systematic empirical work is just beginning.

In conclusion, the Demographic Model described here is best characterized by the side variety of kinds of migration models that can be defined by alternative feedback equations. Clearly, social processes other than migration can be represented by such models, but that is another subject.

Social Distance: A Network Approach – James M. Beshers and Edward O. Laumann

If occupation as structure is viewed as a network, a desirable statistical technique for the measurement of social distance is one which considers all possible paths through the network. Such a statistic is the mean first passage time, from the theory of stationary stochastic processes. Here the technique is applied to some "classic" social mobility data and to associational data. In the mobility data, distance from the top to the bottom of the occupational scale is greatest for the British data, and progressively less for the Danish, U.S. (1910), and U.S. (1940) data, in that order. The behavior of the distance to top professional and semi-professional categories stands out, as does an overall hierarchical effect. In the analysis of the associational data, with a reinterpretation of the statistic, the technique was applied to data on the status of friends, neighbors, and father and father-in-law. A status bias in friendship choice was revealed, and a lesser bias for neighbor's status. (See Beshers and Laumann, Appendix C.)

ADMINS — An Administrative System for the Management of Social Data - Stuart D. McIntosh and David Griffel

ADMINS is a computer-based system for the complex manipulation of social data files and the administration of these manipulated files. ADMINS, as currently operating on CTSS, offers the following capabilities to its users.

1. The Organizer subsystem accepts as input a description of a data file and instructions for transforming the items in the file (these two comprise the 'adform' — short for administrative form), and checks the adform for internal consistency. The Organizer outputs a 'computer-executable' adform. The type of data that can be described in an adform is called heavily structured; i.e., where the location of a code in the record determines the meaning of the code, as opposed to lightly structured data forms (e.g., text flows). The adform has formatting capabilities adequate to describe all computer-readable heavily structured data records (e.g., alphanumeric cards or tape, multi-punched cards, binary data files, etc.). This means any existing computer-readable heavily structured data file may — without any special preparation — be inputted to ADMINS.

2. The Processor subsystem reads the actual data file, searching for discrepancies between adform norms stated with varying degrees of explicitness and the actual data, and transforms the data as instructed by the adform. Various error reports and aggregate reports — both under

interactive control — may be produced, as well as, eventually, an output data file which is in exact correspondence with the 'machine-executable' adform.

3. The File Inverter changes the storage organization from one based on the item to one based on the category of information. As the category record (the protocol in the new organization) contains the normative adform information, a codebook can be generated from the category record which is a description in correspondence with the current state of the data.

4. The Analysis subsystem is very complex, but in summary it allows the following:

1) The construction of indexes; i.e., lists of pointers to items in the file, based on the content of the record. The indexes are named by the user, who may subsequently combine them by name using Boolean operations (e.g., 'and', 'or', etc.) That is, by indexing, the user simulates a restructuring of the file which suits his purpose. The restructuring is organized by the user according to a naming scheme which he assigns to his constructions as he proceeds.

b) The summarization of the file by table construction where the rows and columns of the table are arbitrary indexes and the cells the sizes of the intersections of the indexes. Various statistical tests may be brought to these tables.

c) The analysis system may be used on multi-source data; i.e., a group of files in conceptual relationship. All analysis is done 'in parallel' to these files.

d) The analysis system may be used on multi-level data; i.e., two or more data files at different levels of aggregation with cross-reference tokens in each file relating it to the other file(s).

ADMINS is computer-based (as opposed to people-based) in the sense that the computer has — at the clerical level — a very good idea of what the user is doing and therefore is always watching for various user errors and is ready to provide information on request as well as to retrospectively reference previous work done. The interactive capability of CTSS is fully used in support of intellectual decision-making, especially at the analysis stage.

As a listing of one's data, i.e., a data file describing many data files, is usually also heavily structured, ADMINS may be used (as it is on a current application) to manage one's data base.

ADMINS, as it is developed, will have the following capabilities:

1. The analysis of 'directed graph' type data, where items are linked to other items by ordered relationships; e.g., sociometric data; bibliographic citations between journal articles; kinship relations between data, report, and intermediary files sitting on disk and tape.

2. A lightly structured capability; i.e., an Organizer Processor subsystem capable of taking lighter data forms (e.g., short paragraphs to 'controlled' text).

3. An acquisition capability; i.e., the ability to update an existing data store based on computer inspection of various differing records in the environment.

4. A process-catalogue-based, macro-organizational capability; i.e., giving over to the computer the task of managing — over-time — the hundreds of data, report, and intermediary files users are generating. By 'over-time' we mean the system will relegate — as required — files to lower levels of storage (e.g., tape) yet retain a people-usable retrospective capability. The process catalogue will be based on file content; i.e., the kinship relations existing between data files and support files generated towards a particular purpose.

Models of Turkish Attitudes - Leslie L. Roos, Jr.

Project MAC facilities were utilized in several research projects undertaken or completed during the past year. For my dissertation I used MAC facilities to reweight survey data and test the resulting models against the actual sample survey information. (See Roos, Appendix B.)

In more detail, a model defining respondent-type categories on the basis of sex, level of education, and frequency of newspaper reading was developed to predict the distribution of responses in urban areas in Turkey by means of a weighted average of village input data. By testing the model against available town, city, and metropolitan data, it was possible to gain an idea as to the amount of error likely to result from this procedure; the error was generally less than 2 percent for the prediction of town and city data, and almost 4 percent for the prediction of metropolitan data. Differences in attitudes held by the more educated

metropolitan respondents were identified as being responsible for much of the error in prediction of the metropolitan data from the village input.

Further improvement of the model was attempted through application of the concept of additivity and the use of data from more than one survey. Neither attempt improved model prediction of the urban data. The lack of improvement of model fit as a consequence of combining input for both 1962 and 1963 surveys led to a consideration of reasons for this finding. An analysis of sampling differences and questionnaire content implied that the differences in responses to the 1962 and 1963 items represented actual attitudinal change independent of procedural differences between the two studies.

Finally, a simple model was devised to study relatively long-term changes in village attitudes as a function of change in distribution of village respondent types. The implication of this model was that change in distribution of village attitudes due to respondent-type changes is likely to be very slow over the twenty-year period from 1962 to 1982. On the other hand, the 1962-1963 trend data seemed to indicate that, under certain conditions, village attitudes may change rapidly. By considering political events in the 1962-1963 period, an effort was made to reconcile these differing results.

More recently, Project MAC facilities were used to combine and analyze several surveys of Turkish villagers and bureaucrats. Necessary programming for panel and cohort analysis (techniques permitting the study of change over time) was done for the relevant surveys. Moreover, dictionary lookup techniques were employed to permit the coordination of job history information on bureaucrats with data from the villagers for whom they were in charge. This work is of considerable interest both for development theorists and for students of attitudinal influence; the research will be continued over the next year.

Analysis of the Turkish Peasant Survey - Allan R. Kessler

As part of the continuing analysis of the Turkish Peasant Survey (see Selles and Kessler, MAC Progress Report III), we have prepared our data for input to the ADMINS System. Having utilized ADMINS on a small sample of our respondents, we are now preparing the entire sample for on-line analysis. (See Frey, Appendix C.)

CRISISCOM Simulation - Allan R. Kessler

The CRISISCOM simulation of decision-making during a crisis has been revived in the past year (see Pool, Progress Report II). Continuing

study of the week preceding the outbreak of World War I through simulation of the Kaiser and the Tsar is being undertaken.

Conflict and Consensus in Venezuela - Jose A. S. Michelena and Carlos Domingo

Our objectives in this research program are several:

1. Analysis of survey data using the ADMINS system;

2. Developing programs for survey analysis;

3. Building a computer model of a social system.

The following is a report of the progress to date in achieving the abovementioned goals.

1) Use of ADMINS System. The ADMINS system (being developed at Project MAC by Stuart McIntosh and David Griffel) has been used for the analysis of four of the twenty-eight samples that we are planning to analyze. This work has familiarized us with both the system and the data so that we are now in a position to proceed swiftly toward a final analysis of the samples.

A more concise program in which the information will be reduced to 35 computer words per respondent is being written by Mark Fineman in consultation with Stuart McIntosh and David Griffel. The information required for analysis will be put in a binary format that is ADMINS readable.

2) Testing a Program for Making Analysis Plans. Empirically reducing masses of survey data is one of the most difficult problems an analyst faces. Conceptually defined systems very frequently are not empirically corroborated. The set of programs that we are devising will take a large number of survey variables and arrange them in such a way as to allow the uncovering of the empirical systems underlying the data (i.e., uncovering internal relationships within certain sets of variables as opposed to relationships of one set of variables with other external sets).

For testing purposes we will first select a priori 100 variables out or the 349 contained in the CONVEN survey. This will be done off-line. The next step is computing a matrix of the probabilities of association among each of the 100 variables. For this purpose we will use Chi-square tests at varying levels of significance, or, in the case of numerical variables, the correlation coefficient. This matrix will serve as input to the cluster analysis program which we have already developed. The output will then be a new matrix in which variables will be clustered showing which are subsystem variables and which are

linking variables. On the basis of this, an analysis plan of the samples of university professors and student leaders will be devised and carried out using ADMINS.

3) Programming the VENUTOPIA model. The object is to build a new version of VENUTOPIA using the substantive information and methodological tools prepared in 1) and 2), research of data analysis, and past experiences with the VENUTOPIA first version.

Once the data analysis is completed, the model will be designed in four stages:

i) Make the list of variables and relationships;

ii) Use cluster analysis to find the subsystems and connecting variables;

iii) Program and test each subsystem, adjusting the parameters with the technique developed in 2);

iv) Assemble the subsystems into the total system.

The tests of the model proceed according to the results of the techniques developed in 1). The procedure is:

i) Define the range of parameters and initial values (input space);

ii) Take samples of input space and try them in the model;

iii) Classify the points of input space through analysis of output;

iv) Search for simple relationships between significant input and output subsets.

VENELITE Biodata Analysis - Frank Bonilla and Philip Raup

The ADMINS system is being applied to two main types of data on Venezuelan elites — biographical material and detailed accounts of major transitions (job changes in individual careers). Three principal lines of analysis are being pursued: 1) the construction of complex indexes representing major dimensions of elitehood and social mobility; 2) tracing sequential patterns of movement in occupational and organizational careers, i.e., the details of change or movement in the active lifetime of elite individuals; and 3) efforts to locate difference in time, or across functional spheres, in the modalities of job and organizational ascent. In the last several months, work has concentrated on the organization of the data into the ADMINS format and preliminary indexing and testing of relationships. A second level of analysis of selected aggregate indexes over the several distinct subsets of data is now being initiated.

SCHOOL OF SCIENCE

Computer-Aided Interpretation of High-Resolution Mass Spectra

Neutron Bound and Continuum State-Wave Functions

Nucleon-Nucleon Interaction

Man-Machine Interaction in Meteorology

Exploring the Use of TIP

Demonstrations to Visitors

Use of TIP

Introducing Students to Time-Sharing

Radiative Corrections Programs

Study of Ge(Li) Detectors

Academic Staff

K. Biemann	J. G. Charney	R. I. Matales
S. C. Brown	L. N. Howard	A. Toomre

Non-Academic Research Staff

E. C. Bartels	M. J. Kahne	N. C. Spencer
E. J. Campbell	A. Luehrmann	E. F. Taylor
B. A. Johanson	E. F. Miller	

Research Assistants and other Students

P. V. Fennessey	D. D. Y. Ryu	J. E. Spencer
R. C. Gammill		

Computer-Aided Interpretation of High-Resolution Mass Spectra - Klaus Biemann and Paul Fennessey

The development of techniques routinely permitting the determination of complete high-resolution mass spectrum, i.e., the elemental composition of each ion formed, has considerably increased in both quality and quantity the information available by mass spectrometry. This increase in the complexity and volume of the data has, however, also increased the complexity of the necessary interpretation.

Our problem has been to convert a few hundred lines on a photographic plate, whose distance along the plate is related to mass by the simple equation $m/e = H^2R^2/2V$,* to a list of intensities and masses and then interpret these data, in light of the known fragmentation of organic molecules, to arrive at a structure or possible structures.

The first half of this task, namely the conversion of lines on a plate into accurate masses, was accomplished quite readily using batch-processing techniques. With chain programs we were not only able to do this, but also to carry the processing one step further and have on the final output the elemental composition for all the ions — all this in a two- or three-minute time span.

The next logical step was to see just how far we could program the complete interpretation of a spectrum. This was done using conventional batch-processing techniques with the idea that we could just expand our chain program to the appropriate number of links. However, in the conversion of the present knowledge of the fragmentation processes into a general algorithm for interpretation it soon was apparent that the limiting factors to this approach were core size and, more importantly, time.

While looking for alternate solutions we were introduced to the "time-shared" computer system at Project MAC. This method eliminated the necessity of writing complex "decision tree" type programs, for the interpreter was now able to guide the computer through the desired steps, thus bringing to bear both the knowledge and reasoning power of the operator and the accuracy and speed of the computer on a given set of spectral data.

---

* m = mass of ion in (a.m.u.)
e = charge of an electron
H = magnetic field intensity
R = radian of ion path
V = acceleration voltage

The program itself, which is called INTER, was modeled after the ED and EDL commands, thus allowing the interpreter to choose the sequence of and the restrictions on any set of requests. The programs are all open-ended and the list of available commands has steadily increased to its present level of ten.

INTER is being used for the interpretation of the spectra of various types of organic compounds. By its use, the programs are being changed and altered according to the comments and suggestions of the people routinely using the system.

An example showing the use of a few of the commands is given below, along with a short explanation of each.

Test C99H99N102

In order to deduce the elements present in the compounds, the ions of low mass (i.e., < 100) are tested for N- or O-containing ones, based the fact that a nitrogen- or oxygen-containing compound will produce some small fragments retaining at least one of these atoms.

locate C1, Br, S

The presence of C1, Br, S, or Si is tested by checking the presence of pairs of ions differing by 1.998 ± 0.003 mass units and are of approximate intensity ratios between 1:1 and 30:1.

print top 10

The ions of highest mass in the data are then asked for. From these the most probable molecular ion is chosen by inspection and its elemental composition is tested using the elements found present in steps 1 and 2, in addition to C and H.

call $C99H99N_xO_y$

Once the most probable elemental composition of the compound is thus found, the composition of all ions in the spectrum are calculated with that restriction and summarily presented in the form of "ion-types" with decreasing intensity. An ion-type refers to the ions belonging to the general series $C_n H_{2n+x} N_y O_z \ldots$. Within each type only n varies while x,y,z ... remain constant, i.e., it is a homologous series of a given degree of saturation (also expressed in number of rings and/or double bonds, e.g., 2F refers to a fragment with two rings or double bonds) and heteroatom content. This representation is chosen as the most compact one relating best the characteristics of an ion of interest to the chemist.

Neutron Bound and Continuum State-Wave Functions* - Elmer C. Bartels

In the theoretical nuclear physicists' attempt to explain physical phenomenon he most often chooses a mathematical model with which he may describe the characteristics of the phenomenon. One model for describing the neutron, either bound by the nucleus or scattered off the nucleus, is the Schroedinger Equation which is the second-order linear differential equation

$$\frac{d^2U_{\ell j}(r)}{dr^2} - \frac{\ell(\ell+1)}{r^2} \cdot U_{\ell j}(r) + V_{\ell j}(r)\, U_{\ell j}(r) + K^2\, U_{\ell j}(r) = 0 \qquad (1)$$

where the variable K^2 is defined

$$K^2 = 2 - \frac{m}{\hbar} E \qquad (2)$$

and $V_{\ell j}(r)$ is the potential of the Woods-Saxon type.

If, in the above equation, the energy of the neutron E is negative the neutron is said to be bound and the differential equation must satisfy the boundary condition

$$\underset{r \to \infty}{U_{\ell j}(r)} = k\ r\ H\binom{1}{\ell}(k\ r) \qquad \underset{r \to 0}{U_{\ell j}(o)} = 0 \qquad (3)$$

The value of E which causes this boundary condition to be satisfied in the presence of the potential $V_{\ell j}(r)$ is said to be the eigenvalue of the equation and is the binding energy of the neutron. The associated eigenfunction, often called wave function, when normalized

$$\int_0^\infty U_{\ell j}(r)\quad U_{\ell j}(r)\quad dr = 1 \qquad (4)$$

is useful in further calculations.

When the energy E is positive the neutron is said to be in the continuum, that is, the neutron is scattered from the nucleus, and the wave function is solved subject to the boundary condition that

---

* Research work supported in part by AEC Contract No. A(30-1)-2098.

$$\underset{r \to 0}{U_{\ell j}(r)} = 0$$

$$\underset{r \to \infty}{U_{\ell j}(r)} = k\, r \left[\cos(\delta_{\ell j})\, j_{\ell}(k\, r) - \sin(\delta_{\ell j})\, n_{\ell}(k\, r)\right] \tag{5}$$

where $\delta_{\ell j}$ is the phase shift which is a measure of the interaction between the incident neutron and the target nucleus. The wave function when normalized at infinity as in Eq. (5) may then be used in further calculations.

The code entitled RADIAL on Project MAC CTSS has been written to calculate both the Bound state and Continuum state wave functions as described above and the solutions may be saved internally for use in calculation of radial integrals of the form

$$\int_0^{\infty} U_{\ell j}(r)\; U_{\ell' j'}(r)\; U_{\ell'' j''}(r)\; U_{\ell''' j'''}(r)\; r^n\, dr \tag{6}$$

The code is used often now as it stands and it is envisioned that the addition of a section to take phase shifts over a range of ℓ values and calculate cross sections will be added.

Another capability of the program RADIAL is to calculate solutions to the equation

$$\frac{d^2 U_{n\ell}(r)}{dr^2} - \frac{\ell(\ell+1)\, U_{n\ell}(r)}{r^2} - 2m\ (1/2m\,\omega^2 r^2 - E_{n\ell})$$

$$U_{n\ell}(r) = 0 \tag{7}$$

which are called Harmonic Oscillator Wave functions. The quantity $\omega/2\pi$ is the classical frequency of the oscillation, and E_n is the eigenvalue which takes on discrete values

$$E_n = \left[2n + \ell + 3/2\right]\hbar\,\omega \qquad n = 0, 1, 2 \ldots \tag{8}$$

An exact solution to the Harmonic Oscillator Wave equation may be obtained in the form

$$U_{n\ell}(x) = \left[\frac{2\sqrt{\beta}\quad n!}{\left[\Gamma(n+\ell+\frac{3}{2})\right]3}\right]^{1/2} (\sqrt{x})^{\ell+1}\, e^{-x/2}\, L_n^{\ell+1/2}(x) \tag{9}$$

where $L_n^{\ell+1/2}(2)$ is the Laguerre polynomial and the wave function is normalized

$$\int_0^\infty U_\ell(r) \quad U_\ell(r) \quad dr = 1 \tag{10}$$

These harmonic oscillator wave functions may also be saved internally and used in conjunction with the bound and continuum wave functions to calculate radial integrals.

Further modifications to the RADIAL program will probably include the addition of a wider variety of potentials and possibly the option to specify any potential form during execution time. The program is written to give a conversation "question-and-answer" session. to make usage by the non-programmer easy. Many branching possibilities are provided in the program to allow different potential parameters, energies, and quantuum numbers, plus the options to save wave functions for later use or to print wave functions.

Nucleon-Nucleon Interaction - Nancy C. Spencer

The present work consists of finding a nucleon-nucleon potential consistent with experiment and the boson exchange theory of nuclear forces. Research is conducted under AEC Contract A(30-1)-2098. (An article covering the theory and results of this work will be published shortly by Professors E. Lomon and H. Feshbach of the M.I.T. Physics Department.)

Rho, omega and eta resonances were added to the one- and two-pion exchange potentials. With the aid of CTSS, the form of the potential could be modified easily and quickly and the results of the change seen. The theoretical nucleon-nucleon potential of Cottingham and Vinh Mau was also programmed and investigated.

The main objective of the program is to find a minimum set of parameters which characterize the potential and give a "best fit" to the available nucleon-nucleon data. Some of these parameters are highly correlated, so it is necessary to find which parameters are independent and to devise a method whereby these parameters can be varied most effectively in searching for a good fit. Off-line search time should be greatly shortened by first using CTSS to find a first approximation to a good solution and by examining the correlations between parameters. Then the parameter space surrounding this first approximation can be examined more extensively to optimize the solution.

Other programs related to the nucleon-nucleon interaction are also being written, tested and debugged.

Man-Machine Interaction in Meteorology - Julie G. Charney, Peter Webster, and Robert C. Gammill

The object of our research is the formulation of meteorological models which would allow meaningful interaction to take place between meteorologist and model. This is difficult by the fact that meteorological models tend to be large and complex, both in the size of the program and in the amount and variety of data. Furthermore, teletype terminals are inadequate for printing output when used in the manner of a line printer. Thus, meteorological models must be programmed in entirely new ways when used under time-sharing than they would need to be under batch-processing.

During 1966-67 our research centered on several models of the atmosphere which treat the three-dimensional distribution of various quantities in the atmosphere as one-dimensional fields and arrays of Fourier components. This affects considerable reduction in the amount of data required by the model without removing physical effects which are of interest. A further advantage is that the MAP system (see MAC-TR-24) with its prepackaged set of graphical display, data manipulation, and analysis routines allows us to create input data, analyze output, and display results in a very flexible manner.

It must be emphasized that it has been possible for us to work with the model under a man-machine environment due to careful restriction of the size and the form of the model, and due to the mathematical tools available through MAP. Work is now proceeding on various graphical tools which may allow us to work with somewhat less restricted models in the future.

Basically, the model described above allows, in a simplified manner, for the interaction of the mid-latitude wave perturbations with the model zonally symmetric component of the flow from the equator to the pole and for the effect of the release of heat of condensation on the low latitude circulation. Of major interest is the formation of an "Intertropical Convergency Zone" near the equator and its development or dissipation under the influence of various forcings, such as the release of latent heat and the interaction of the mid-latitude perturbations.

Exploring the Use of TIP - Sanborn C. Brown

We would like to know whether the Project TIP program for information retrieval from current physics literature has real educational value for graduate students. To determine how useful this program would be, we have set up a console in one of the graduate student's offices of the Physics Department (Room 4-332), available primarily for teaching assistants. For several hours a week we have had available

to the graduate students using this area an instructor to teach the graduate students how to use the program. This duty has been shared between Edward M. Mattison of the Project TIP Staff and myself. No conclusions have yet been reached about the usefulness of this project, though it has been quite heavily used as indicated by the exhaustion of alloted time before the end of the month at those times when the teaching assistants have been using their offices, namely, on shifts 1 and 2. Further information will have to be gathered before an evaluation of this effort can be made.

Demonstrations to Visitors - Elizabeth J. Campbell

We gave console demonstrations to several physicists visiting the Laboratory for Nuclear Science. For example, we discussed and demonstrated the use of time-sharing in the field of physics with a visitor to our PEPR project, Dr. R. Rigopoulos from the Centire Demokritos, Athens, Greece.

Use of TIP - Elaine F. Miller

The physicists and students within our laboratory have made extensive use of TIP, the M.I.T. library's literature-searching routine. Work was supported by AEC Contract No. A(30-1)-2098. It has been a great help to them in their research. Since most of these people were not familiar with either programming or time-sharing I wrote an informal memo containing the basic time-sharing commands as well as instructions for using TIP. We found that this memo, in conjunction with demonstrations at the console, enabled non-programmers to effectively use the system within an extremely short period of time.

Introducing Students to Time-Sharing - Elaine F. Miller

During the course of the year, members of the Electron Interaction Group found that writing short programs on time-sharing served a dual purpose. These routines served not only to yield specific answers to specific problems but also served to introduce graduate students within the group to the capabilities and techniques of time sharing. The programs include a routine which fits p-p scattering data and routines used in various phases of checking the programs involved in our experiment at the Cambridge Electron Accelerator. The experimental programs are, in part, concerned with elastic scattering of electrons by helium 4 nuclei.

Radiative Corrections Programs - Elaine F. Miller

The facilities of Project MAC have been used in experimenting with modifications to the set of programs we have for dealing with problems associated with the radiative degradation of electron-scattering spectra.

The original sequence of programs was written to take experimentally determined electron spectra and unfold them from the undesired radiative effects. To determine the corrected inelastic cross section for scattering of primary electrons we must determine the uncorrected cross section over a triangular region. In general, the correction programs will operate in this triangle with a substantially smaller mesh than it is practical to employ when taking data. Our first program accordingly takes a limited amount of experimental data and by interpolation techniques generates the full mass information on which subsequent analysis programs operate. Another program computes elastic and inelastic yields and thus serves as a source of predictions of data. As the set of programs continues to expand, decisions about methods of modification become increasingly more difficult. We have found time-sharing to be a useful tool in developing and checking these modifications. In the future we will continue to use time-sharing to develop new approaches to changes in our basic program.

<u>Study of GE(Li) Detectors</u> - James E. Spencer

We are presently using the LNS time-sharing console to determine how one can best construct and use germanium lithium-drifted radiation detectors for experiments in nuclear physics. Although these detectors represent one of the more significant advances of the past decade in the field of nuclear research, they have yet to be fully understood or exploited. Undoubtedly, a major cause of this lies in present technical limitations associated with their fabrication process. Nevertheless, there is a large amount of work of rather general application which can be performed which is independent of the above limitation.

Our investigation concerns some of the "kinematic" aspects of the overall problem of how one can best use such detectors. For instance, we would like to know what is the optimum size and shape of detector for a given set of experimental conditions. This involves numerically solving integrals representing the total detection efficiency, which is proportional to the integrated pulse height spectrum. However, to obtain the peak efficiency there is no such simple procedure. Since this is really what we are interested in optimizing, together with the peak resolution, we are writing a program to simulate the detector using Monte Carlo techniques. Time-sharing seems well suited to such a problem, since many modifications to the basic program may easily be tested at the console.

SLOAN SCHOOL OF MANAGEMENT

On-Line Incremental Simulation

Array Manipulation

Project Scheduling Research

DYNAMO Compiler

Computer-Aided Teaching System (CATS)

Computer-Aided Diagnosis

Marketing Model Construction

Marketing Information Systems

A Model of Personnel Flow

Computer Utility Cost and Price Determination

On-Line Statistical Analysis and Simulation

Console-Operated Statistical Routines

DATANAL: An Interpretive Language for On-Line Data Analysis

Multi-stage Manufacturing Simulator

Scheduling Production in the Dynamic Closed Job-Shop

Resource Constrained Project Scheduling

A Teaching and Operational Aid for Applications of Statistical Decision Theory

A Technique for Dynamic Alteration of Model Structures

Academic Staff

S. S. Alexander	W. R. Fey	M. M. Jones
A. E. Amstutz	J. W. Forrester	D. A. Kolb
D. C. Carrol	M. Greenberger	J. D. C. Little
P. H. Cootner	W. H. Gruber	C. V. Swanson
W. B. Crowston	M. Haire	L. S. White
D. E. Farrar	D. M. Holland	

Non-Academic Research Staff

J. R. Miller, III	A. L. Pugh, III	D. M. Straight, Jr.

Research Assistants and Other Students

D. A. Belfer	H. R. Johnston, Jr.	W. D. Reinhardt
R. E. Boyatzis	F. Kamijo	T. W. Schwenke
J. R. Brown	L. S. Kanodia	L. L. Selwyn
J. Casher	M. Karr	J. A. Sutton
D. S. Diamond	J. L. Linderman	R. C. Thurber
M. M. Gold	L. M. Lodish	R. P. Tinkleman
G. A. Gorry, Jr.	D. F. Manchester	M. A. Trozzi
R. S. Green	J. H. Morris, Jr.	P. M. Walker
J. J. Grimes	J. L. Openshaw	R. D. Wright
B. J. Huffman	E. V. Pedersen	
T. J. R. Johnson	I. Plotkin	

On-Line Incremental Simulation - Martin Greenberger and Malcolm M. Jones

Availability of the Project MAC time-sharing system for the past several years has made possible the implementation of an experimental on-line simulation system called OPS-3.* Development of the MULTICS second-generation time sharing system at Project MAC has encouraged us to attempt an improved version of this simulation system, called OPS-4, which has a cleaner conceptual structure and the hope of better operating performance. The new system includes features of Multics, and uses experience gained from OPS-3. The motivation and characteristics of the system were described in a paper presented to the IFIP Conference on Simulation Languages in Oslo, Norway, on May 22-26, 1967. The following is a summary of the important features of the OPS-4 system. (See Jones, Appendix B.)

1. PL/I is the basic language and provides a general algebraic and data-handling facility.

2. The system is specifically designed to encourage a user to build a model incrementally, and test a partial model before all the pieces have been completed.

3. The "world view" encompasses material- and flow-oriented models, machine- and entity-oriented models, or models combining both views. Either activities or events may be represented.

4. Special data types — known as sets, queues, and tables — are available in addition to the normal data types of PL/I. There is no limit to the number or size of any data types.

5. There are statements for the generation of random deviates from the more widely-used distributions.

6. Communication among program elements and variables in the model can be controlled, but is not restricted.

7. Restructuring of the data base does not require recompilation of procedures. The normal mode of execution is interpretive.

8. The status of a model may be saved at any point during a simulation by executing a single statement.

---

*On-Line Computation and Simulation: The OPS-3 System, M.I.T. Press, Cambridge, Mass., August 1965.

9. Reinitializing data bases and resetting system time is easy. Thus it is convenient to restart a simulation or execute a chain of simulation runs. A model may be restored to a previously saved state or rolled back to a previous point in simulated time.

10. The user has flexible controls to specify the exact order in which events are executed.

11. No important part of the system is hidden from the user. He has direct access to (and ability to modify) every key element of the simulation from his console.

12. Extensive debugging and tracing features are available, and they are easy to use.

13. The structure of a model may be modified without recompilation.

14. Flexible means are provided for specifying the starting and stopping points or duration of a simulation run.

15. Individual components of the model can be independently tested, even if embedded in larger modules.

16. The user can interrupt a model at any point during the execution phase, redirect its path, examine and change the values of variables, then continue the simulation from the point of interruption.

17. In unusual situations (e.g., an attempt to run time backwards), the user is given the benefit of the doubt and the simulation is not interrupted. However, a flag is set.

18. There are comprehensive facilities for collecting statistics.

19. Only the structure and mode of initial data inputs in procedures need be declared by the user. The structure and mode of most data objects resulting from a computation are inferred by the rules of the computation.

20. Immediate on-line diagnostic explanations are provided when an error is detected during the running of a model. The length and detail of these messages is under user control.

21. Debugged portions of a model may be compiled and run at full speed. Interpretive execution is used for sections of programs not yet checked out.

Array Manipulation - James H. Morris, Jr.

The array manipulation facilities of the OPS-3 system were revised and extended. The conventions for element-by-element operations on arrays were generalized and the printing of arrays was improved.

Designation of array subparts — If A denotes a matrix then

A [i, j] denotes the element in the ith row and jth column,
A [i, *] or A [i] denotes the ith row,
A [*, j] denotes the jth column,
$A[i_1 \ldots i_k, j_1 \ldots j_e]$ denotes a k by e submatrix.

These subscript conventions apply to vectors and cubes as well.

Implicit iteration — Normal algebraic expressions, such as found in MAD or FORTRAN, can be used to perform computations on arrays. For example, if A is a cube, B a matrix, C a vector, and D a scalar, typing

```
SET  A = B + C * D
```

will have the same effect as the following FORTRAN program segment:

```
      DO 10  I = 1 , N
      DO 10  J = 1 , M
      DO 10  K = 1 , L
10    A (I, J, K) = B (J, K) + C (K) * D
```

Where N is the depth of A, M is the number of rows in A and B, and L is the number of columns in A, B, and C.

In general, if two variables of different dimensionality (e.g., a vector and a matrix) are combined with a scalar operation, one or both are expanded in the dimensions they lack.

Other operations — Matrix multiplication and transposition are also available. A' denotes a column (row) vector if A is a row (column) vector, denotes the transpose of A if A is a matrix, and denotes the transpose of each ply of A if A is a cube.

The product of two matrices is obtained by typing

A .M. B

while the inner product of two vectors is typed as

A .M. B'

Further, the sum of all the elements in a vector is

A .M. 1

Efficiency — Matrix operations involving arrays of over one hundred elements (e.g., a 10 x 10 matrix) take less time than the corresponding operations in compiled MAD programs.

Project Scheduling Research - Wallace B. Crowston

Project scheduling research may be categorized by the types of constraints found in the models developed. The CPM algorithm determines minimum completion time of a project given precedence constraints. A class of resource allocation models assumes precedence constraints on available resources. Heuristic routines are used to sequence the jobs under the limited resource constraint in an attempt to minimize project length. A recent article* on the time-cost tradeoff problem suggests that discrete alternative methods may exist for performing some of the jobs in a project and that interdependency constraints may hold between these alternatives. The set of alternatives that minimizes the sum of job cost and cost of the completion date is selected by integer programming or branch and bound techniques.

The current research deals with projects that contain discrete alternatives as well as interdependency resource and precedence constraints. The goal is to select that set of alternatives which, when optimally sequenced under the resource constraint, minimizes the sum of job cost and cost of project completion date. However, it is not possible to solve such problems optimally, since for any given set of alternatives selected it is not computationally possible to find the optimal sequence. Since heuristic methods are used to solve the sequencing problem, only an approximate evaluation can be made for any set of alternatives chosen.

An interactive program package is being developed to approach this problem. Essentially it consists of an integer programming routine that handles the interdependence constraints and a heuristic routine which handles the precedence and resource constraints. The routines are linked by an interactive segment in which the operator evaluates a heuristic solution to the sequencing problem and sets pseudo-shadow prices on the discrete alternatives in the programming problem. The process iterates until no further improvements are found. Although this particular study is in the context of project scheduling, the routines developed will have significance for a wide range of design problems.

---

*Crowston, W. B., and G. L. Thompson, "Decision CPM: A Method for Simultaneous Planning, Scheduling and Control of Projects", Operations Research, Vol. 15, No. 3.

<u>DYNAMO Compiler</u> - Alexander L. Pugh III, Jay W. Forrester, Carl V. Swanson, Willard R. Fey, Erik V. Pedersen, and Donald A. Belfer.

This past spring DYNAMO II has operated on time-sharing. This compiler was designed with three objectives in mind:

1. To make it easier to convert DYNAMO to different machines. Toward this end we wrote DYNAMO in AED-O. AED has been an excellent language in which to write a compiler and it should be a good language for converting to other machines.

2. To include a general algebraic translator patterned after the techniques of Samelson and Bauer. The techniques have been extended to provide clear error messages.

3. To improve error recovery, so that more errors can be detected in any one compilation, and so that the model can be run should the errors be trivial.

Our experience this spring, though limited, has indicated that the second and third objectives have been achieved. This summer we begin translating DYNAMO to S/360 and will discover whether the first objective has also been achieved.

In the future we hope to add two more features to DYNAMO II: to permit arrays (or boxcars, as they were called in DYNAMO I) and to permit the user to create macros, a feature that is new in DYNAMO II.

<u>Computer-Aided Teaching System (CATS)</u> - Leon S. White and H. Russell Johnston

The objective of this project is the development of a software package to provide facility for constructing programmed-learning presentations of various subjects on time-sharing computers. The package permits the teacher to place the teaching program on file through use of the EDIT command (ED. TAPE. xx where x= tape number) to create a dummy tape, using nine mnemonic codes as follows:

1. QU - causes the material on succeeding lines to be treated as a question for the student to answer

2. CA - material following this code is the preferred answer to the preceding question

3. CB - delineates alternative correct answers

4. WA - indicates the most significant wrong answer

5. WB - specifies further anticipated wrong answers

6. TY - provides the facility to respond to each of the above anticipated answers (both correct and wrong answer)

7. UN - provides a "catch-all" response to a wrong answer from the student which has not been anticipated by the instructor. This code must follow each QU group.

8. SU - enables the system to present a summary or re-cap at the end of the teaching program

9. END - denotes the end of the program.

Once the program is placed on tape, it must be transformed into a format which can be used interactively by the teaching routine. This is achieved by issuing the command LOADGO FREERD XREADQ which will request the problem number. This routine will create another dummy tape, numbered 50 below the problem number provided by the instructor (e.g., if the instructor inputs "51" as the problem number, the student will request problem number "1" when he uses the system).

For the student to use the system, he need merely log in and issue the command LOADGO STUDEN XREADQ. He will be asked what problem he wishes to work. After inputting this number, the tape which has been created with that number will be called and instruction will commence.

Future extensions of the system should provide greater branching flexibility and a scoring capability.

Computer-Aided Diagnosis - G. Anthony Gorry

This Ph.D. research deals with a model diagnostic problem and a computer program designed to deal with this problem. The model diagnostic problem is an abstract problem. A major contention is that this problem subsumes the principal features of a number of ostensibly different real diagnostic problems including certain problems of medical diagnosis and the diagnosis of machine failures. A second major contention is that strategies for the solution of the model diagnostic problem can be formulated in terms sufficiently explicit to permit their incorporation in a computer program.

The model diagnostic problem assumes that the system being diagnosed (e.g., a person, or machine) is in one of a finite number of known states. Tests can be performed at some cost to discover attributes of the system, for example signs or symptoms in medical diagnosis. The current state of the system is to be deduced from the observed attributes and past experience with similar systems. In the model, this experience is represented principally in terms of probabilities (e.g., the conditional probability of a certain attribute given the system state).

The statement of the model diagnostic problem requires that the diagnostician also account for the cost of various misdiagnoses. In particular, for each pair of states i and j, the cost of misdiagnosing state j

as state i, l_{ij}, is given. Thus the diagnostician must balance the cost of performing additional tests against the expected reduction in the cost of misdiagnosis. This requirement suggests the value of sequential diagnosis.

A computer program was developed to solve the model diagnostic problem. It consists of 1) an inference function which is based on a Bayesian analysis of attributes and includes a flexible way of dealing with non-independent attributes, 2) a pattern-sorting function which allows the program to detect irrelevant attributes and patterns of attributes corresponding to two different system states, and 3) a test-selection function which employs various heuristics to select good tests for the user of the program to perform on the system under consideration. The diagnostic program is specialized for a particular problem by providing it with the appropriate experience. The program is embedded in an environment (set of programs) which facilitates the study of various diagnostic strategies.

The diagnostic program was implemented on the time-sharing system at Project MAC. It was applied to two medical problems; the diagnosis of congenital heart disease, and the diagnosis of primary bone tumors. The results obtained here suggest 1) that a computer program can be of considerable value as a diagnostic tool, and 2) that it is quite advantageous for such a program to perform sequential diagnosis as it interacts with the user. (See Gorry, Appendix B.)

Marketing Model Construction - John D. C. Little and Leonard H. Lodish

A number of marketing models have been built or are under construction. The computer side of the research emphasizes the advantages of interactive computer capability either in the model building or in the model use or both. Of particular interest are steps that bring the model closer to the operating manager.

Work previously reported on a geographical model of an urban automobile market has been largely completed. Several outside demonstrations of the interactive use of the model have been given and a paper on the model is in the process of publication.

Current work has concentrated on a model for advertising media selection and its implementation as an interactive computer program. The model may be described briefly as follows: the customer population is divided into market segments. Each segment has its own sales potential and media habits. A media schedule consists of insertions in media vehicles (e.g., a full-color page in a particular magazine or a 60-second spot on a specific TV show). An insertion brings about exposures in the various market segments. However, people tend to forget, and so the

retained exposure level decays with time in the absence of new exposures. The anticipated sales to a market segment increase with exposure level but with diminishing returns.

The task of preparing a media schedule that maximizes anticipated sales for a given advertising budget can be formulated as a mathematical programming problem. The problem can be solved by exact techniques or, more efficiently, but without a guarantee of optimality, by heuristic programming. An interactive English language program has been developed for the model and its optimization. A number of demonstrations have been given and a fair amount of interest has been generated among advertising men.

Marketing Information Systems - John D. C. Little and Daniel S. Diamond

A program of research has started in the area of marketing information systems. One aspect of this is the determination of the need and probable use of interactive capability in such systems. A prototype system has been developed and is being studied and modified.

A Model of Personnel Flow - Mason Haire

The project was designed to explore the feasibility of a formal statement of personnel flow in an organization over time. Inputs are the optional interventions of the organization (e.g., training, pay, promotion, etc.). Outcomes are changing transitional probabilities with respect to position in the organization. These probabilities depend on states of the individuals, and consequently, sums of these states are evaluations of the organization at times and places.

During the year, the model was formalized and programmed and now provides an interactive system for simulation and analysis. The next step is to provide coordinating definitions for an operational test.

The model was presented to the Visiting Committee of the Sloan School, using a teletypewriter console and closed-circuit TV, so that the group could interact with the system, varying values and adding parameters as states developed over time. It has also been presented at the General Electric and Sun Oil companies, where we hope to find operational tests.

Computer Utility Cost and Price Determination - Lee L. Selwyn

The objective of this study is to develop a methodology for economic evaluation of the operations of computer utility systems, with a view toward techniques of internal management and external customer relationships.

To this end, effort has been directed, in the past year, toward the development of techniques of resource management within the Project MAC environment. The author has held the responsibility for computer usage allocation and administration on the IBM 7094 system, under CTSS operation, for two of the user groups at Project MAC, comprising approximately 125 individual users. For each of these two user groups, a different administrative system has been established.

The Sloan School of Management group (user group 17) has been the subject of an experimental pricing system, whereby users have been able to obtain allocations of computer resources by spending units of an artificial currency which is distributed among them. The resources so controlled are processor time, divided into five "shifts", comprising various portions of the calendar week, and disk storage space. The five time shifts are 1) weekdays, 8:00 to 5:00 p.m., 2) weekdays, 5:00 p.m. to midnight, 3) weekdays, midnight to 8:00 a.m., 4) weekends, 8:00 a.m. Saturday to 8:00 a.m. Monday, and 5) Foreground-Initiated Background, actually a low-priority class of service, rather than a division of the calendar week.

As administrator, the author acted to alter prices of commodities so as to equate the available supply of each resource to its demand. A forthcoming improvement to the system would provide an automatic price-setting mechanism, based upon a statistically-determined prediction of overall monthly demand. In its initial months, the operation of this experiment led to several important, though admittedly preliminary conclusions. It has been demonstrated that, even under conditions of non-profit organizations and artificial currencies, it is indeed possible to provide effective resource administration by the imposition of transfer prices. Users do tend to maximize their individual utilities, such that, by providing monetary trade-offs among the various system resources, users did indeed begin to work in the otherwise more unpopular times of the week. Also, by providing a direct means of conversion of disk tracks to machine time, it was found that users tended to be more efficient in their use of the storage medium. As a result, it has been possible to make more of the Sloan School's allocation of resources available to its faculty and students. Increased efficiency of usage has been made possible through another important means — that of providing assistance and information to individual users on the various techniques available to them within the CTSS environment. This has been a major activity of the author in his capacity as group administrator.

One important limitation of the pricing experiment is that it has been applied to a relatively small group of users, perhaps less than 40 persons, so that the existence of relative interdependencies among users (in the economic sense) has tended to reduce the effectiveness of the pricing mechanism as a control device. It is hoped that, at some point

in the future, an experiment of this nature will be made using a larger group of individual users, so as to minimize their interdependence and thereby maximize the effectiveness of a free market mechanism.

To facilitate the process of resource administration, the author established, for the Computer Systems Research Group (user group 6) a resource management system that provides for direct responsibility of sub-project leaders for the specific amount of computer time and disk space consumed by researchers and system programmers under their control. (This is perhaps analogous to the Project-wide practice of dividing up the responsibility for resource allocation by means of relatively small user groups, such as the two under discussion here.) Although this system was not part of the study under discussion, it did serve to demonstrate yet another distinct method of resource management. Although there are no prices and currencies, sub-project management has been most effective in allocating the relatively scarce computer resources to those individuals whose work, considering specifics of timing and scheduling and completion of larger objectives, was most important to the Multics development efforts. The fact that frequent changes in resources allocations are made by sub-project leaders attests to the usefulness of this tool.

In the coming year, the completion of a Ph. D. dissertation on this subject is anticipated. The thesis will consider such questions as the nature of the computer utility industry, the nature of the individual firm in this industry and its various economic (and perhaps social) objectives, and the nature of the various aspects of management of the computer utility firm.

<u>On-Line Statistical Analysis and Simulation</u> - David A. Kolb and Donald A. Belfer

Two areas of time-sharing development are being pursued in conjunction with research on feedback and personal change in groups. The first is the practical usage of an on-line interactive system for statistical analysis of a large data base. (See Miller, this section.) The system will be oriented primarily to the early phase of social science research, in which the direction and formulation of specific research hypotheses are unclear.

A second area is the development of on-line simulation techniques integrated with access to the data base. This would facilitate the building and testing and mathematical simulation models by providing the ability to estimate parameters and perform statistical validation of simulation results in an interactive fashion.

Aspects of other data-analysis being developed concurrently have been investigated, with attempts to integrate them with existing simulation systems. A prototype simulation model has been built from a large data base that was already processed to some extent by off-line routines. We are now attempting to program the necessary routines within the context of on-line access to the data base. This will allow further development of the simulation techniques.

Console-Operated Statistical Routines - James R. Miller, III

The library of statistical routines, under development since 1964, has been further expanded. Additions to the library over the past year include:

1. A two-sample Chi-square test of homogeneity;
2. A binomial test on the signs of differences between matched pairs of either ordinal or cardinal simple observations;
3. A Wilcoxon matched-pairs signed Ranks test;
4. A T-test on matched pairs of cardinal sample observations;
5. Tests for symmetry and normality in sample data.

The entire library of routines was then revised to permit a less restrictive format for typing in data and the ability to correct erroneous inputs. (See MAC-TR-40, Appendix D.)

DATANAL: An Interpretive Language for On-Line Data Analysis - James R. Miller, III

An interpretive language for data analysis has been created. The name of this language is DATANAL. It has been designed to facilitate:

1. Analysis of any kind of empirical data collected in any context;
2. On-line conversational interaction between a user and his data through the medium of a time-shared computer;
3. Two-way communication between user and computer in English (i.e., no additional programming languages need be learned);
4. Immediate usability by individuals relatively naive with respect to computers and their idiosyncracies.

A library of console-operated statistical routines being developed by the same author since 1964 has been incorporated within DATANAL so that requests for such analyses may be issued in English and executed immediately.

Multi-stage Manufacturing Simulator - James J. Grimes

This research is concerned with the design construction, and operation of a three-stage, interactive production simulation model. The basic model consists of three operating stages separated by buffer inventories. Each stage has the capability of assuming varying production rates over time, breaking down at random intervals, and having normally distributed repair times. In addition, each stage has a quality rate associated with it, the quality deteriorating over time as determined by a transition matrix. The operation of each stage is also affected by the availability of maintenance.

Each buffer inventory is examined to determine when it will either runout or overflow, the overflow level being variable. Either condition results in a shutdown of the appropriate stage. Levels may be set to provide warnings of either a runout or overflow condition. The stages are restarted whenever a set amount of items have either been removed or added to the inventory depending upon the condition for shutdown.

A filing mechanism operates at set time intervals to record the values of selected parameters. Routines to display this information are to be constructed.

Currently the design and programming for the basic model have been 90 percent completed, and debugging of the remaining system is in process. A plotting mechanism for displaying the recorded parameter values has been completed and is in operation.

The next step, after completing the debugging, will be to gain operating experience with the model and modifying it based upon this experience. There are plans to construct additional I/O routines to allow the user to interrupt and modify the system values during the simulation run. It is also expected that the system will be expanded to provide computer decision rules.

Scheduling Production in the Dynamic Closed Job-Shop - Lalit S. Kanodia

This research effort has been directed towards integrating batch or lot size determination, sequencing, and the management of finished goods inventory. The environment in which this has been attempted is the job-shop; hence the term "Dynamic Closed Job-Shop".

The dynamic closed job-shop is identical to the job-shop scheduling problem as defined in the literature, except for the single difference that the shop manufactures only a limited number of products for which it processes repeat orders. This difference introduces two problems which would not have existed otherwise, viz. "how should jobs be batched while processing?" and "how should finished goods inventory be managed?".

The problem-solving methodology being used consists of analytic and heuristic reasoning. This led to the creation of what appeared to be good alternative policies by which production in such a shop could be scheduled. These policies consisted of a combination of job-releasing, batch-size determination, and job-sequencing rules.

The relative performance of these policies has been tested by simulation. The simulator used for this purpose has been coded in FORTRAN 2 and consists of over 2000 instructions.

The research has evolved in strategic and tactical plans for validating hypotheses by experimentation involving simulation as the principle apparatus. The results obtained are capable of being generalized to encompass most simulation experiments.

The performance of the shop as regards schedule performance can be significantly altered by alternative methods of batching. Considerable evidence exists to show that batch sizes should not be permitted to vary across the machine centers for the same job. The policy recommended permits the variation of batch sizes across jobs. It has been demonstrated that the hybrid job-releasing rule, which would release a batch of jobs whenever the number sold equals the batch size processed by the shop, is suited to this shop.

Among the sequencing rules tested, the LPD family of rules is recommended by virtue of its superior performance and robustness. They outperformed the SST rule and equalled the performance of the LPD/T rules. Simplicity has been used as the criterion in recommending the LPD rules over the LPD/T rules.

The rules employ a current and global data base. This can be construed as evidence in support of on-line real-time data-processing. It is expected that this research will bring us much closer to the realm of operationality in the field of scheduling than we have been hitherto.

This research effort has yielded a Ph.D. thesis for the author at the Sloan School of Management. The research effort was also supported in part by a Doctoral fellowship awarded by the Ford Foundation. A plant of the Ford Motor Company provided the motivation for the model used in this research. (See Kanodia, Appendix B.)

Resource Constrained Project Scheduling - Thomas J. R. Johnson

A "branch-and-bound" algorithm for the resource-constrained project scheduling problem has been developed and programmed. Briefly, the problem is to find a schedule for a known set of tasks that minimizes the time required to complete all work, while observing a complex set of logical task-precedence requirements and stated limits on resource use in each time period.

If each problem is viewed as a non-stochastic sequential decision process, the "solution space" can, in theory, be completely described by the end points of a huge "decision tree". Each intermediate decision point can be thought of as a "partial schedule". The basic computational problem is to find an optimal schedule without exploring more than a small fraction of the decision tree.

The algorithmic scheme is to explore a sequential "route" through the decision tree until the resulting partial schedule is an improved complete schedule, or is clearly nonoptimal (that is, if a minimum bound on all complete schedules containing this "partial" is greater than, or equal to, an existing solution or if the partial is demonstrably dominated by another partial). In both cases, the algorithm returns to explore other routes and continues until all unbounded, undominated partial schedules have been examined. Judicious selection of alternatives at each decision point leads to good upper bounds rapidly, and list-structured data permits swift machine computations.

The computer program (in FORTRAN IV) solves in less than a minute most problems containing less than 50 tasks. However, some problems must be terminated prematurely after exceeding available storage or reasonable time allotments (e.g., 10 minutes). As problem "size" approaches 100 tasks, nearly all problems have to be halted prior to assured optimality.

Since most practical problems involve a more complex situation than that modeled (and more tasks to be scheduled), the main thrust of the research is to provide an excellent vehicle for heuristic insight to the problem. Given an optimal choice at each decision point, we can examine how it differed from its alternatives. One such difference, "conflict" when all unscheduled tasks are scheduled at their "earliest start dates", is under study at present.

<u>A Teaching and Operational Aid for Applications of Statistical Decision Theory</u> - Mark A. Trozzi

The object of this research project was to develop a prototype computer program, to be used in a teleprocessing environment, for application to problems in the field of statistical decision theory. It was required that the program perform two functions:

1) Provide computer support which, by freeing the student from the time-consuming (and error-prone) task of evaluating decision trees, would allow the learner to focus his attention on the methods of statistical decision theory. In this manner and use, the program would be a self-teaching device or teaching machine.

2) Provide an operational aid to the student who has formulated a problem using statistical decision theory concepts and would like a computer system to handle the arithmetic details of the solution. In this use, the program would function as a calculation aid.

The program called the Decision-Making Aid (DMA) system does more than merely perform arithmetic calculations. The DMA has been designed to lead the user through all the operations necessary to the solution of the problem, execute error checks on the data which is input, provide for error recovery, and evaluate the utility function of the user on request. With the option to use either an Expected Monetary Value (EMV) or Risk Averse (RA) utility function, the decision maker becomes aware of the effects of his attitude towards risk on the solution to his problem. If requested, the DMA will solve the user's problem with his utility function and then display the EMV solution for reference.

The current system can provide solutions to decision-tree-formatted problems with a maximum of 2401 unique branches, and permits no data storage within the system. Input/output is handled through a teletype or typewriter console. The suggested areas for future study are:

1) Expand the size of the input arrays to permit the system to handle larger problems.

2) Provide for storage of data on user problems within the system.

3) Explore the advantages and disadvantages of video display units.

(For other suggestions and further documentation of the system, see Trozzi, Appendix B.)

<u>A Technique for Dynamic Alteration of Model Structures</u> - Fumihiko Kamijo

The structure of a model and the specification of activities are treated unchangeably in the execution of a model, which are written by conventional simulation languages. However, events vary dynamically in the real world. A dynamic representation of a model is necessary for detailed analysis.

A study to formulate dynamic alterations in a simulation system has been made. The new system handles the structure of a model as data separated from the event-scheduling process. The dynamic model structure is represented as a tree, which is an entity used uniquely in this system. In the course of a simulation, transactions walk along nodes

(event nodes) of the event tree. A transaction is a collection of scheduling system attributes and local data (parameter and history). It also carries local linking information. A dynamic loading system is proposed to maintain the global linking information for programs and data.

With the above tools, an improved image can be obtained of the dynamic behavior of a model.

TECHNICAL INFORMATION PROGRAM

Project TIP

A. Systems

B. Subsystems

C. Components

Physics Data Revisions

Academic Staff

S. C. Brown	M. M. Kessler

Non-Academic Research Staff

J. P. Casey	W. D. Mathews	P. M. Sheehan
T. F. Dempsey	K. D. Rude	W. A. Solomon
E. A. Dole		

Research Assistants and other Students

A. S. Gevins	L. H. Morton	D. S. Breindel
T. M. Higgins	A. Pawlikowski	H. Helava
L. K. Marquardt	B. L. Zimmerman	K. Sills

Project TIP - Meyer M. Kessler

The stated purpose of Project TIP has been two-faceted: first, the design, construction, and prototype operation of an experimental communication system for the technical community and, second, the use of this prototype as a test bed for ideas and components to provide operations experience for evaluation. The intent has been to exploit modern computer technology in the cause of scientific communication and to set up a working model of a system that will demonstrate the potential promise of the engineering contribution to this field.

Our goal throughout this period had been to engage in activities of wide application rather than to offer a massive service in any one area. The following functions have been developed and tested, with work supported in part by the National Science Foundation under Grant GN-589.

1. The basic TIP library of close to 100,000 articles from physics journals is in machine-usable form. Of these, 30,000 are on the disk, the rest on magnetic tape. This library grows at the rate of some 1200 articles per month.

2. The TIP search subsystem allows a variety of console operations to be performed on the above library for purposes of search and retrieval.

3. A book and several review articles have been published which were based on TIP literature search (see references 1, 2, 3).

4. The TIP command is in constant use from some twenty consoles each week.

5. A console has been installed in a room used by physics graduate students who are now writing theses. The use of this facility is being monitored and observed.

---

[1] Brown, Sanborn C., Basic Data of Plasma Physics, 1966; Library of Congress No. 67-15601, M.I.T. Press, Cambridge, Massachusetts, 1967

[2] Ashburn, Edward V., ed., Laser Literature, A Permuted Bibliography, 1958-1966; 2 vols., Western Periodicals Company, North Hollywood, California, 1967

[3] Ashburn, Edward V., and Bela A. Lengyel, "Bibliography of the Open Literature of Lasers, VI.", Journal of the Optical Society of America, 57, 1, January 1967, pp. 119-148

6. We have instituted a weekly notification system informing M.I.T. physicists when one of their papers is cited.

7. Social scientists and historians have used TIP to study the flow and dynamics of scientific literature.

8. Several students are using the facilities of TIP for research papers and theses.

9. TIP subsystems are being used by several administrative offices at the Institute as an aid in their managerial functions.

10. TIP subsystems collect, process, and prepare for publication a catalog and inventory of M.I.T. journal and periodical holdings.

This work has produced greater understanding about such matters as file organization, the handling and manipulating of structured lists, the organization of large data banks and the design of interactive languages. The generalized TIP system is a very flexible tool that can handle structured data in general, not only scientific information. As such, it is applicable to library catalogs, personnel files, financial and managerial information, etc. These extensions are now being explored and investigated.

Our original goal of prototype design and operation is now more or less accomplished. For the next two or three years, we have in mind the following course of action:

1. We should continue our present research and development program, including the transition of TIP into the new computer environment.

2 Responsible administrative arrangements must be evolved to accept TIP functions when they reach the service level.

3. We must train people who can exploit the operational capabilities that we have already achieved.

4. We have to extend our investigations into areas where legal, social, and human problems predominate.

5. We should integrate our work with other research and teaching activities on the campus.

The computer, as a logical machine, has application not only as a bookkeeper but also as a correlator of recorded knowledge and a discoverer of relations between isolated bits of information whose record, although available, is so dispersed that the integrated pattern does not emerge. Knowledge, like energy, must be gathered into a concentrated package in order to be useful. It must be brought into focus in space and

time. Only then can human imagination play on it and acquire insight. This is a gleam and a conjecture, but the record of our experimental system leads us to believe that it is a direction that we should speculate on.

Our report is in three descriptive sections: systems, subsystems, and components. A system is a unified process involving many procedures, not all of them necessarily mechanized. Subsystems are equivalent to CTSS commands, and are command-level programs used as system building blocks. Components are the software subroutines from which subsystems are made.

A. SYSTEMS

Input - Timothy F. Dempsey

The data base for the TIP system is typed on paper-tape recording machines. The full ASCII character set is available. This data is loaded onto the disk through the photoelectric reader on the PDP-7 at Project MAC. Large volumes of data can be efficiently introduced in this way. Interpretation of erase, kill, and escape characters, and canonicalization of overstrikes is also done as part of the input system. None of the convenient features of on-line data creation are lost, yet the system costs less than one-tenth as much as console input would cost. Code conversions from paper tape to ASCII are table driven, and it is possible to construct tables for special punching configurations with a minimum of difficulty. Conventions for splicing, packaging, and labeling the paper tapes have made control of input a routine operation.

Inventorying and Certification - William D. Mathews

Nearly 15,000 items, each having fifty or more fields of information, are introduced into the TIP data library each year. Maintaining controls to insure quality production is a difficult task. Item inventory lists indicate the date of creation and size of each item. File inventories list the items in each file and allow the orderly editing of these items.

In checking new items for the data base, three levels of certification are recognized: formal, structural, and textual. Formal certification of an item means that it has been properly named and can be retrieved by its expected name. Structural certification means that the item has the proper number and configuration of fields. The most complete assurance to the user of the accuracy of TIP-searchable data is textual certification, a guarantee of correct spelling, punctuation, and typography throughout. Extremely high reliability of this sort is impossible to achieve without much repetitive, prohibitively costly, hand labor. As the best alternative, one cycle of proofreading and correction of each item is performed. This produces data of moderately high reliability.

TIP-Initiated Alerting - Kenneth D. Rude

One of the experimental services Project TIP is engaged in is to notify members of the M.I.T. community automatically whenever one of their papers, published in a physics journal, is cited by another paper also published in a physics journal. These notifications are known as MAC-TIPS and are derived as a by-product of the normal TIP data-input scheme. Their production is quite automatic.

As new physics data are loaded onto the disk and introduced into the TIP library, each item is inspected in detail against profiles of seventy M.I.T. authors. The profiles consist of notations about papers written by these authors together with any other pertinent requests. Whenever any of these papers are cited in the new input literature, a notice is sent to the author giving bibliographic information about the citing article. In addition, the system is automatically updated by a search against the same weekly input for any new papers published by these same M.I.T. authors.

Topical Bibliographies and Statistical Techniques - William D. Mathews

Faced with the task of producing automatic bibliographies on many topics in the physics field, we have developed several statistical techniques. Given certain relevant articles, it is possible to develop a statistical profile of this set of papers and use that profile for retrieval of further articles. For example, we may look for all articles with the word "laser" in the title. These articles are then examined and their citations statistically tabulated. The result is a profile of commonly cited articles in the laser field. These citations may then be used in a new retrieval request. Articles citing the laser literature but not having "laser" in the title may thus be retrieved. Properly handled, this kind of reaction stabilizes after a few cycles. The strong points of title, author, or citation retrieval complement each other in this technique.

Tape - Kenneth D. Rude

File storage requirements for the TIP programming group are such that a substantial portion of the total required must be on magnetic tape. Since the TIP data files grow at a rate of several thousand disk tracks a year, it appears likely that a requirement for magnetic tape storage will exist for quite some time.

An effort has been made to simplify and regularize the use of magnetic tape. The names of files which are to be transferred from disk to tape are placed in a special file, known as a directory. Each person in the group may create a number of his own directories. Group projects also have directories.

Initially, the several disk files mentioned in a directory are written on a scratch tape in a single file for each directory. Later, the person in the group responsible for tape handling transfers the files on the scratch tape onto permanent storage tapes.

The external directories are the focal point of this system. As disk files are written on tape, the line in the directory file corresponding to the file just written is modified to include the length of the file stated in words. Thus, files can be combined on the disk or on tape to save space, and to be searched, etc., as a single file without the disadvantages of ARCHIV, which inserts control information between the files combined.

To split the file previously combined, the directory is again read and causes the file to be split and named according to the file names and word counts found in the directory. Comments may be placed in the directory by beginning a line with an asterisk. We use comments to indicate the tape label and file number, the data created, and the approximate number of disk tracks the split files would require.

The directories always reside on the disk. Essentially, then, this is an embryonic form of multilevel file system. Should a user wish to release a tape file he no longer needs, he enters the name of the directory file associated with the file to be released into a small deletion file. This is periodically consulted by the tape librarian and the files the directories point to are purged. In addition to the relative ease of tape handling which this system affords, it has saved us quite a bit of time — both computer and human.

<u>Current Serials and Journals (CSAJ)</u> - Patricia M. Sheehan and Brenda L. Zimmerman

The CSAJ system, established in 1966 as a prototype for automation of an entire library system, created a disk catalog for approximately 6500 serials and journals in the M.I.T. libraries. In 1967, the system was expanded to include controlled updating processes and to extend the options available to console users.

At present, information concerning new subscriptions, recatalogued titles, relocated issues, etc., is entered on a newly designed form which serves to initiate processing in the appropriate areas of the Technical Services Department of the Libraries and also as a direct source for computer input.

The initial programs subject the input to exhaustive validity testing, edit the data, and generate related changes. If the journal is new, the program assigns a permanent-file digit ID rather than the complete title. In this case, the program consults a computer-created "ID-Sequence Directory" and assigns a Sequencing Number based on the last

relative position of the journal in the disk catalog. The input is then sequenced by title, in the case of new journals, or by Sequencing Number, in the case of old journals. The validated input is used to update the master disk files. Additional validation and generation of data is done at this point. As a final updating step, a new "ID-Sequencing Directory" is created. The updated file may be used at any time for hard-copy catalogs, statistical studies, or retrieval requests. Hard-copy output is printed either on a standard 1401 or on a special 1401 with an upper- and lower-case print chain. For larger volume production, the output is converted to film and either Xeroxed or converted to surface offset plates for press publication.

The series of runs producing hard-copy catalogs offer the console user the following options:

1. The catalog may include entries from all M.I.T. Libraries, from particular divisional libraries, or from specialized collections. (Non-M.I.T. entries may be treated as libraries or collections.)

2. The catalog may be limited to journals covering subjects identified by call number classification or by title text.

3. The catalog may be limited to entries added, changed, or deleted since the last catalog published by the M.I.T. Press.

4. The entries may be identified by the computer on output either explicitly by label or implicitly by position.

5. Editing parameters may be set for single or double columns, all upper- or lower-case, number of lines on page, number of characters in line, etc.

The statistical studies serve three general purposes:

1. Analyses to be used for library planning such as volume or duplicated titles among libraries and the relative weight of such duplications for the given library or group of libraries.

2. Matrices identifying and ranking catalog components.

3. Control figures for TIP, M.I.T., and professional studies.

Many of the programs must necessarily be geared directly to details and conditions peculiar to libraries and to the M.I.T. Libraries in particular. Programs which are of more general use include the following:

UPDATE Package

American Library Association (ALA) filing rules have subjective criteria and demand sequencing by grammatical word, inter-filing of words with their abbreviated forms or alternate spellings, separation by hierarchy of works by corporate and by individual authors, and elimination of punctuation marks on the basis of their identity together with their textual value.

As mentioned in our 1966 report, a computer analysis was made of the difference in results produced by traditional computer sorting techniques and by manual methods following ALA rules. In published material on library automation, the solution has been to require extensive manual tagging within input text.

Our package eliminates such tagging by the following process:

(a) The input form shows the title exactly as catalogued.

(b) The program retains the catalogued title but creates, as necessary, a "sort title" in the following way:

   (1) Initial articles in six languages are identified and eliminated.

   (2) Abbreviations are expanded.

   (3) Punctuation marks are classified by means of test analyses as "sorting" or "non-sorting" and the latter eliminated.

   (4) All numerics are padded to a standard six-character size.

   (5) Bit patterns with new sequencing values are assigned to all surviving alphabetics, numerics, blanks, and punctuation marks.

ID-SEQUENCE Directory

This set permits the system to bypass sort-editing and textual sorts for titles previously entered on disk. The program uses the permanent ID number to find in the current directory the most recent disk position of the title.

UPPER-LOWER Package

This group of programs translates disk text stored in upper-case to upper- and lower-case. To do so, it analyzes each grammatical word. Acronyms are made completely upper-case; articles, prepositions, and conjunctions following the first word of text, completely lower-case; all other words follow standard capitalization rules.

FIT Package

This group prevents the splitting of words between output lines, prevents the splitting of all data for one title between columns, and produces double-column output of variable length records and by printing each double-column line of text as soon as the right-hand half has been prepared.

FOOTNOTE Package

Updating input may include text which, when encountered in other records, should be emphasized by an asterisk and an explanatory footnote. Both the asterisks and the footnotes are generated automatically.

Among the users of the system in the last year were the Birla (India) Institute of Technology and Science — catalogs covering general science and mathematics; M.I.T. Press — Current Serials and Journals in the M.I.T. Libraries; University of Oregon — edited magnetic tape to serve as nucleus for their library automation; M.I.T. departments — catalogs covering Life Science, Mathematics, etc.; M.I.T. library administration — statistics, data for expansion studies, analyses of call number revisions, internal worksheets, reports to U. S. Office of Education and to the Association of Research Libraries.

B. SUBSYSTEMS

These subsystems exist as saved files and may be used privately by any user of CTSS. The TIP subsystem exists as a command. The others may be accessed by linking.

TIP - William D. Mathews

The TIP command is the generalized retrieval subsystem developed by Project TIP. It is an interactive system, allowing for considerable conversation with the user. Retrieval can be performed on itemized data according to complex selection criteria. Selection is based on fields having a specific content or falling within a range of values. Part of the interactive language is user defined and can be easily modified to reflect the nuances of the user's data. Selected items may be intricately formatted for output purposes.

The output may be on the console or into a file and may be in a form which is immediately searchable for further retrievals. Alternatively it may be formatted so as to be reducible, for restructuring and retrieval, or printable for book production.

Interaction involving TIP, Sort, and Reduce can dramatically alter the contents and arrangement of a file. This subsystem has been carefully tuned for speed and efficiency.

EDIT - Lewis H. Morton

EDIT is a string manipulating language for the CTSS time-sharing system at M.I.T. It is a highly interactive interpreter, with programming capability. The language provides both context sensitive and context free manipulation of strings. Strings are dealt with in terms of symbolic names. The workspace may be implicitly structured by the contents of the string contained in it. Input and request may be read from the console, or ASCII coded files. Output may be to the disk or console. Error messages are complete and understandable. The modularity of the language allows the more sophisticated user to modify the system easily.

While not as efficient as TIP for retrieval purposes, EDIT does not require itemized data bases and therefore has wider possibilities.

SORT - William D. Mathews

It is possible to organize original or derived data bases in many different orders.

The Sort package arranges TIP data on any fields the user wishes. Author and citation indices have been created on the Physics data. Subject indices, dictionaries, and personnel files have been similarly maintained by other users.

MERGE - William D. Mathews

Maintenance of sorted data bases can be done with the Merge subsystem. Options allow for updating as well as the more traditional merging processes. Selection can also be accomplished by Merge. Thus it acts in a limited way as a retrieval function. Sort and Merge together provide the basic file maintenance for TIP.

REDUCE - William D. Mathews

Because of the inherent limitations in the instruction set in the IBM 7094, data to be used with the TIP systems is preprocessed to convert it from an ASCII representation to a list structure. Reduce structures the data internally into items and fields.

EXPAND - William D. Mathews

Expand reverses the Reduce process. It is possible to use Reduce and Expand repetitively to reorganize and restructure the information in TIP files. Fields may be redefined as items or items as fields. Options allow for limiting formatting of the output.

INDUCE - Timothy F. Dempsey

There are a number of discrete stages in the processing of TIP bibliographic data from the Flexowriter form to the final "searchable" form. The programs necessary to process this data have been subsumed by a larger subsystem called "INDUCE".

There are currently four programs and two "pseudo-programs" in INDUCE. The four programs are: CVFILE, REFORM, REDUCE, and REVERT. The pseudo-programs are: FORM and MAKE. FORM is simply a combination of CVFILE and REFORM, while MAKE combines REDUCE and REVERT.

CVFILE may be used to convert the coding system of the characters of a file from any 9-bit code to any other or to the two CTSS coding systems and vice-versa. On user option, ESCAPE, KILL and ERASE characters are recognized and the appropriate action is taken. CVFILE has a wider application than just the processing of TIP bibliographic data.

This is not true, however, for the other three programs. They are peculiar to the TIP style of data.

REFORM presents the raw data in a formatted, easy-to-read print style.

REDUCE creates searchable data from either the reformed or raw data. It differs from the REDUCE subsystem in that there is special processing for bibliographic data.

REVERT converts the output from the above REDUCE to the form consistent with the command TIP.

TCHECK - Edward M. Mattison

The program TCHECK provides a console-initiated automatic process for checking the presence of all fields and, therefore, all articles in given issues. TCHECK compares the contents of one or more issues with special "check files" which are intended to be completely accurate representations of the issue contents.

A check file consists of journal, volume, and issue identification, and a series of numbers giving the first pages of all articles in the issue. They are made from Xeroxed copies of each issue's table of contents. The check file production proceeds concurrently with text processing so that the check file and text are ready for comparison at the same time. After disk input, the check file is proofread and corrected on-line until it is known to be accurate with a high degree of confidence. The comparison by TCHECK of text and check file produces on the console notice of gross errors, such as missing files or misnumbered issues. TCHECK

also creates a file containing page numbers of all articles with missing fields together with a statement of which fields are missing with each article.

RUN - Kenneth D. Rude

Many of the TIP programs are command-level programs which are called from the console or by using the FIB monitor. Since it is expensive to store a large number of SAVED files, or to recreate them each time they are to be used, the RUN package was evolved to allow more efficient SAVED file creation and storage.

In most SAVED files the main (or utility) program is unique to that file. The subroutines which are called by the utility programs are usually common to all TIP programs. Therefore, much of the disk track space used to store separate SAVED files consists of identical copies of the TIP internal subroutines. The RUN package circumvents this waste by creating a single SAVED file which contains all of the utility programs together with the subroutines necessary to load successfully.

Since RUN is intended to be merely a convenient way of storing SAVED files, it has been constructed to require only a small alteration in the writing of utility programs.

To use one of the TIP utility routines, for example REDUCE, the user would type on the console, "Resume RUN REDUCE Arg 1 ... Arg n". The program RUN in the file RUN SAVED reads the (MOVIE TABLE) and returns to free storage all memory taken up by the utility programs except that used by REDUCE. Finally, the program exits to REDUCE. Some of the utility programs are written to be called either through the RUN package or as subroutines.

By using the SAVED file system just described, Program TIP has experienced 75 per cent saving in disk space to store SAVED files.

CSAJ - Brenda L. Zimmermann

During the past year, programs have been written which provide tabulations of statistics on the Current Serials and Journals data which are useful to the project and to the libraries. One program sorts entries and tabulates title totals, first by library and stacking symbol and, second, by classification. Another program determines the longest and average length of each data field with the call number field broken down by type. Other programs print those titles which have cross references, accompanied by the references, and count duplicate titles held by various combinations of M.I.T. libraries. Recently finished programs will slightly modify the format of the CSAJ master file and will create a directory for easier updating and correction of those files.

C. COMPONENTS

TIP Component Packages - All the TIP subsystems are involved in text manipulation and depend heavily on the components or subroutines we have developed for this purpose. The task of identifying and specifying components was very large. Over 140 functions are defined in our general component file. These functions are divided into six packages.

TIPI/O - Timothy F. Dempsey

A new B-core input-output package has been developed for dealing with ASCII files. At the present time, the package can handle up to ten files at once in either a single-buffered or double-buffered mode. The buffers may be of any size which is a multiple of 1728 characters.

This package is considerably faster than older B-core packages for reading files since there is a minimum of data-movement. The user is given a pointer to the data in the buffers with the understanding that the data will remain until the next call for that file.

Opening and closing of files is taken care of automatically by the package, so the user need not worry about that part of housekeeping. Reading and writing need not occur in sequential fashion if the files are on the disk. The package may be told to read from or write into a specific byte number or into a byte relative to the next one to be read or written.

STRING - William D. Mathews

The STRING package manipulates, evaluates and transforms strings of text. For example, SPLIT breaks a string into its constituents and UNIFY re-unites such a fragmented string. COMP compares strings; MATCH looks for a substring in another string. SORT arranges tables of string pointers, and LOOK does a binary look-up on such a table. There are many other entry points to this component package.

LIST - Lewis H. Morton

A List is an open-ended structure maintained in free storage by the LIST Package. Cells, or elements of cells, may be appended to the LIST or written over old cells. As cells are appended, they are given an identifying sequential number. Every list contains an index set to the current cell. This index may be used to identify a cell for input or output and can be reset or stepped. Within lists, cells or parts of cells may be retrieved singly or in multiples, while lists may be deleted or truncated.

A second symbolic device enters here and is called a "Virtual Image" of a list. This consists of a sequence of pointers to cells in a real List, which are sorted into a different order. This sorting is done on a key in each cell which is defined at the time of sort, and which may be any word or portion of a word in the cell.

CORE - Lewis H. Morton

The efficient dynamic management of Core storage is one of the most important problems in our environment.

Free storage space for strings may be reserved and released by using TAKE and GIVE. The CORE package will also insert a string into a collection, returning a pointer to it or will store a pointer in a list which is referenced by name and transaction number, or will combine these functions and return a name and number after storing a string. Deletion calls, in the package, require only the name, and act on the entire collection or list at the same time. The illusion of a rudimentary two-dimensional addressing scheme may be maintained.

UTIL - William D. Mathews

A number of components for conversion and those string functions returning only values are included in the UTIL package. Such functions calculate lengths of strings or distances between edges of strings but do not alter or inspect the actual contents of the strings. Conversions from ASCII to BCD or from decimal to octal are also handled here.

DEBUG - Lewis H. Morton

The TIP debugging tools include modified versions of FAPDBG, an off-line feature for large debugging output called PLUG, and a program timing routine called TIME. The modifications to FAPDBG include 1) multiple, conditional breakpoints, 2) new output modes, including one based on the distance from program load point, and 3) a search feature to find specified words in core. Several small bugs in the CTSS FAPDBG have been corrected.

It is also possible to execute any CTSS command from this version of FAPDBG. The core image of a debugging session may thus be saved for possible roll-back.

PLUG traps calls to WRFLX, WRFLXA, and RDFLXA and diverts the complete interaction into a disc file for large debugging sessions.

TIME allows interpretive execution and counting of FAP instructions. It can apply a test following every executed instruction, and call a debugging subroutine when a criterion is reached (such as a core-location changing contents). Execution with TIME is about 25 times slower than normal.

Physics Data Revisions - Sanborn C. Brown

The computer-based revision* of the book Basic Data of Plasma Physics, 1966 (see Appendix C), served as a successful test of the feasibility of the TIP program. This project was sponsored by the United States Atomic Energy Commission under Contract AT(30-1)-1842. The references to the data are in machine-readable form rather than standard bibliographical form, so that keeping the book up-to-date may be done by the machine on a citation basis. Sufficient time has not yet elapsed since the publication of this book to know whether this citation revision will be useful or not, but the new literature is being kept track of by means of the TIP searches. This is accomplished by means of a file called TITLE UPDATE, which contains all the useful categories used in the production of the book and serves as a FIND command in searching the new literature as it is added to the TIP library.†

---

* Based on the Basic Data of Plasma Physics, The Technology Press, 1959.

† As a direct outcome of Professor Brown's work on the TIP system, he is serving as Chairman of the Advisory Committee to the American Institute of Physics on its Information Program, and in this role has spent a fair amount of time studying the broader implications of TIP and TIP-like programs on the national scene with regard to journal information retrieval problems.

APPENDICES

Appendix A
Projeet MAC Memoranda

Appendix B
M.I.T. Thesis

Appendix C
External Publication

Appendix D
Project MAC Technical Reports

APPENDIX A

PROJECT MAC MEMORANDA

MEMORANDUM MAC-M-No.	SUBJECT	AUTHOR	DATE
314	DSR, An On-line Data Storage and Retrieval	A. Armenti	8/15/66
317	AED Flash 31: FOCL - Frame Oriented Command Language (9442-M-172)	W. D. Maurer	7/14/66
318	New Storage Package (9442-M-173)	D. T. Ross	7/15/66
319-1	AED Flash 32-1: Tracing Subroutine STOPAT (9442-M-174-1)	B. L. Wolman	8/09/66
320	SIDES 21 (AIP No. 104-2)	R. Greenblatt J. Holloway D. Sordillo	8/01/66
321	AED Flash 33: The "FEATURES Feature"	D. T. Ross C. Feldmann	8/15/66
322	Proposal for an Inertial Pen as a Computer Input Device (794-M-7)	D. Haring	8/16/66
323	A Step-by-Step Computer Solution of Three Problems in Non-numerical Analysis (AIP No. 101)	W. A. Martin	7/66
324	An Input Macro for TECO (AIP No. 103-102)	D. Eastlake	9/66
325	AED Flash 34: Available AED System Macros (9442-M-177)	C. Feldmann R. Lynn	9/14/66
326	Music Playing on the PDP-6 (AIP No. 103)	D. Sordillo	8/22/66
327	Symbolic Integration - II (AIP No. 97A)	J. Moses	10/13/66
328	Figure Boundary Description Routines for the PDP-6 Vision Project (AIP No. 104-5)	J. White	9/66
329	A Primitive Control P Feature (AIP No. 103-2)	D. Eastlake	10/66
330	AED Flash 35: Line-Assemble Package (9442-M-179)	P. T. Ladd	10/14/66
331	AED Flash 36: The ASEMBL Package (9442-M-180)	B. L. Wolman	10/28/66
332	Summer Vision Programs (AIP No. 104-6)	L. Lamport	10/66
333	SCPLOT BIN (AIP No. 103-3)	D. Sordillo	10/66
334	CHAR PLOT (AIP No. 103-4)	D. Sordillo	10/66
335	A Description of the CNTOUR Program (AIP No. 104-7)	L. Krakauer	11/66

MEMORANDUM MAC-M-No.	SUBJECT	AUTHOR	DATE
336	Operation and Programming Manual for the ARDS-I Experimental Dataphone-Driven Remote Storage-Tube Display	T. R. Cheek J. E. Ward D. Thornhill	11/21/66
337	Second AED Technical Meeting (9442-M-182)	D. T. Ross	12/07/66
338	IOT Instructions for Special Devices Connected to the DEC Type 338 Display Computer at Project MAC	D. J. Edwards	11/23/66
339	Current Operating and File Systems for MAC 338 Display Computer	T. P. Skinner	12/01/66
340	Telephone Extensions for Dataphones, Teletypes, and 1050's (CC 230-11)	M. Solomita	12/08/66
341	User's Guide to OS/8	T. Skinner	12/14/66
342	A Primitive Recognizer of Figures in Scene (AIP No. 119)	A. Guzmán	1/67
343	AED Flash 37: The Alarm Package (9442-M-185)	A. Mills C. Feldmann	1/31/67
344	AED Flash 38: Delayed Merge Program (9442-M-186)	B. L. Wolman	2/15/67
345	A Quick, Fail-Safe Procedure for Determining Whether the GCD of Two Polynomials is 1 (AIP No. 126)	J. Moses	3/67
346	A Miscellany of CONVERT Programming (AIP No. 130)	H. McIntosh A. Guzmán	4/67
347	Canonic Systems and their Application to Programming Languages	J. Donovan H. Ledgard	4/28/67
349	A Data Storage Structure for On-Line, Multiplexed Information Storage and Retrieval Systems	A. Bushkin	5/67
350	AED Flash 39: RWORD Package (70429-M-189)	S. Ackley W. Johnson J. Porter	5/29/67
351	Message Format and Protocol for Inter-Computer Communication	A. Bhushan R. H. Stotz	6/16/67

APPENDIX B

M. I. T. THESES

Auslander, D. M., Analysis of Networks of Wavelike Transmission Elements, Department of Mechanical Engineering, Ph.D. Thesis, August 1966

Bernhardt, L. J., A Ten-MC Binary Rate Multiplier for Computer Line Generation, Department of Electrical Engineering, S.B. Thesis, June 1967

Blount, F. T., Design of a Low-Cost Computer Graphical Input Device, Department of Electrical Engineering, S.B. Thesis, June 1967

Brody, W. R., Computer Display of Three-Dimensional Scalar Functions, Department of Electrical Engineering, S.M. Thesis, August 1966

Bushkin, A. A., A Data Storage Structure for On-Line, Multiplexed Information Storage and Retrieval Systems, Department of Electrical Engineering, S.M. Thesis, June 1967 (See also MAC-M-349, Appendix A.)

Chan, S-C., A Nonlinear Analysis of Framed Structures, Department of Civil Engineering, S.M. Thesis, January 1967

Deitel, H. M., Monitoring Remote Terminals in a Multiplexed Computer System, Department of Electrical Engineering, S.B. Thesis, May 1967

Domercq, R. J., A Machine-Aided Thesaurus Generation Systems, M.S. Thesis, September 1967

Edelberg, M., The Dual Approach to Threshold Decomposition of Boolean Functions, Department of Electrical Engineering, S. M. Thesis, June 1967

Falk, G., An Assumed Form Realization for Sequential Machines, Department of Electrical Engineering, S.M. and E.E. Thesis, June 1967

Fenichel, R. R., An On-Line System for Algebraic Manipulation, Harvard University, Applied Mathematics, Ph.D. Thesis, July 1966 (See also MAC-TR-35, Appendix D.)

Gorry, G. A., A System for Computer-Aided Diagnosis, Sloan School of Management, Ph.D. Thesis, June 1967

Grimes, J. J., The Design and Operation of a Three-Stage Flow Shop Simulation Model, Sloan School of Management, M.S. Thesis, June 1967

Guzmán, A., Some Aspects of Pattern Recognition by Computer, Department of Electrical Engineering, M.S. Thesis, February 1967 (See also MAC-TR-37, Appendix D.)

Haccoun, D., Simulated Communication with Sequential Decoding and Phase Estimation, Department of Electrical Engineering, S.M. Thesis, August 1966

Hall, E. J., STRUDL Member Selection Procedures, Department of Civil Engineering, S.M. Thesis, June 1967

Henke, W. L., Dynamic Articulatory Model of Speech Production Using Computer Simulation, Department of Electrical Engineering, Ph.D. Thesis, September 1966

Hronik, E. J., Techniques for Efficient Communication Between Remote Consoles and Multi-Access Computers, Department of Electrical Engineering, S.M. Thesis, August 1966

Hsu, W. L., The Computer as an Aid in the Teaching of Bond Graphs, Department of Mechanical Engineering, S.B. Thesis, June 1967

Johnson, T. J. R., An Algorithm for the Resource Constrained Project Scheduling Problem, Sloan School of Management, Ph. D. Thesis, July 1967

Jones, M. M., Incremental Simulation on a Time-Shared Computer, Sloan School of Management, Ph. D. Thesis, February 1967

Kamijo, F., A Technique for Dynamic Alteration of Model Structures, Sloan School of Management, S. M. Thesis, June 1967

Kanodia, L. S., Hueristic Sequencing of Jobs and Batch Size Determination in the Dynamic Closed Job Shop, Sloan School of Management, Ph. D. Thesis, June 1967

Levitt, B. K., High Resolution Beam Pen, Department of Electrical Engineering, S. M. Thesis, February 1967

Llewelyn-Jones, D. T., Far Infra-Red Plasma Diagnostics, Department of Physics, Ph. D. Thesis, January 1967

Malhotra, A., Asynchronous Control of Computer Operations, Sloan School of Management, S. M. Thesis, February 1967

Martin, R. L., Studies in Feedback Shift-Register Synthesis, Department of Electrical Engineering, S. M. Thesis, May 1967

Martin, W. A., Symbolic Mathematical Laboratory, Department of Electrical Engineering, Ph. D. Thesis, January 1967 (See also MAC-TR-36, Appendix D.)

Norton, L. M., ADEPT: A Heuristic Program for Proving Theorems of Group Theory, Department of Mathematics, Ph. D. Thesis, August 1966 (See also MAC-TR-33, Appendix D.)

Patil, S. S., An Abstract Parallel-Processing System, Department of Electrical Engineering, S. M. Thesis, June 1967

Radzikowski, H. A., An Extended Mathematical Programming System, Department of Metallurgy, S.B. Thesis, June 1967

Roos, L. L., Jr., Models of Attitudes in a Developing Country; The Turkish Case, Department of Political Science, Ph. D. Thesis, September 1966

Sibley, E. H., The Computer as Symbol Manipulator and Evaluation for Engineers, Department of Mechanical Engineering, D.Sc. Thesis, February 1967

Slosberg, D. H., On-Line Computer Tools for Feedback Shift-Register Synthesis, Department of Electrical Engineering, S. M. Thesis, July 1966

Smith, M. G., Computer Location of Plasma Dispersion Function Multiple Roots, Department of Electrical Engineering, S. M. Thesis, September 1966

Stoper, A. E., Vision During Pursuit Movement: The Role of Oculomotor Information, Brandeis University, Department of Psychology, Ph. D. Thesis, May 1967

Stuntz, S. C., Inputting Process to an Augmented Library Catalog - An Efficiency Study, Department of Electrical Engineering, B.S. Thesis, June 1967

Trozzi, M. A., A Teaching and Operational Aid for Applications of Statistical Decision Theory, Sloan School of Management, S. M. Thesis, June 1967

Van Horn, E.C., Computer Design for Asynchronously Reproducible Multi-processing, Department of Electrical Engineering, Ph.D. Thesis, September 1966 (See also MAC-TR-34, Appendix D.)

Vasser, E.R., Audio-Coupled Receiver Demodulator for Telephone-Transmitted Digital Data, Department of Electrical Engineering, S.B. Thesis, June 1967

Wessler, B.D., TracD, A Graphic Programming Language, Department of Electrical Engineering, S.M. Thesis, June 1967

Willems, Y.D., Synthesis of Logical Functions with Restricted Threshold Elements, Department of Electrical Engineering, S.M. and E.E. Thesis, June 1967

Wright, D., Sequential Decoding with a Fading Dispersive Channel, Department of Electrical Engineering, S.M. and E.E. Thesis, June 1967

Yochelson, J.C., Multics LISP, Department of Electrical Engineering, S.B. Thesis, June 1967

APPENDIX C

EXTERNAL PUBLICATION

Allen, J. L., and G. D. Bernard, "Superposition Optics - A New Theory", M. I. T. R. L. E. Quarterly Progress Report No. 86, July 1967

Bartsch, R. R., "Beam-Plasma Discharge: System D", M. I. T. R. L. E. Quarterly Progress Report No. 84, January 1967, pp. 142-144

Bartsch, R. R., "Operation with 'Asco' Valve Gas Injections", M. I. T. R. L. E. Quarterly Progress Report No. 83, October 1966, pp. 68-72

Bartsch, R. R., "Operation with Marshall Valve Gas Injection", M. I. T. R. L. E. Quarterly Progress Report No. 83, October 1966, pp. 65-67

Benenfeld, A. R., Gurley, E. J., and J. E. Rust, Cataloging Manual, M. I. T. E. S. L., Technical Report ESL-TM-303, February 1967

Bernard, G. D., "Current Distributions on Dipole-Excited Equiangular Spiral Planes", International Scientific Radio Union (U. R. S. I.), Stanford, California, December 1966

Bernard, G. D., "Research Objectives", M. I. T. R. L. E. Quarterly Progress Report No. 84, January 1967, p. 121

Bernard, G. D., and W. H. Miller, "Interference Filters in the Corneas of Diptera", M. I. T. R. L. E. Quarterly Progress Report No. 86, July 1967, pp. 109-122

Beshers, J. M., "Computer Models of Social Processes: The Case of Migration", Population Association of America, April 1967

Beshers, J. M., "The Design of an On-Line, Large-File, Social Data System", International Symposium on Mathematical and Computational Methods in the Social Sciences, Rome, July 4-8, 1966

Beshers, J. M. and E. O. Laumann, "Social Distance: A Network Approach", American Sociological Review, M. I. T. and the Joint Center for Urban Studies, University of Michigan, Vol. 32, No. 2, April 1967

Bobroff, D. L., and H. A. Haus, "Impulse Response of Active Coupled Wave Systems", Journal of Applied Physics, January 1967

Bobrow, D. G., and D. L. Murphy, "Structure of a LISP System Using Two-Level Storage", Communications of the ACM, Vol. 10, No. 3, March 1967

Brackett, J. W., and R. Kaplow, "X-Y Graph Production and Manipulation", Proceedings of DECUS Spring Symposium, 1967

Brown, S. C., Basic Data of Plasma Physics, M. I. T. Press, Second Edition (revised), 1967

Brown, S. C., "Computer Programmed Basic Data of Plasma Physics", M. I. T. R. L. E. Quarterly Progress Report No. 85, April 1967, pp. 71-72

Brueck, S. R. J., and A. Bers, "High-Frequency Electron-Photon Interactions in a Magnetic Field", M. I. T. R. L. E. Quarterly Progress Report No. 82, October 1966, pp. 72-76

Chou, S-L., and A. Bers, "Dispersion Relations for Beam-Plasma Interactions", M. I. T. R. L. E. Quarterly Progress Report No. 85, April 1967, pp. 168-177

Coons, S. A., "The Uses of Computers in Technology", Scientific American, vol. 215, no. 3, September 1966

Davis, H. S., "Development of Private Library to be Searched by TIP", M.I.T. R.L.E. Quarterly Progress Report, No. 84, January 1967, pp. 359-361

Davis, J.A., "Beam-Plasma Interaction with a Plasma Density Gradient Along the Beam", Physics of Fluids, research note, 1967

Davis, J.A., "Computer Simulation of the Beam-Plasma Discharge", M.I.T. R.L.E. Quarterly Progress Report No. 82, July 1966, pp. 146-151

Davis, J.A., "Sheet Models of the Beam-Plasma Discharge with Plasma Density Gradients Along the Beam", M.I.T. R.L.E. Quarterly Progress Report No. 84, January 1967, pp. 145-148

Davis, J.A., "Sheet Models of the Beam-Plasma Discharge with Plasma Density Gradients Along the Beam", M.I.T. R.L.E. Quarterly Progress Report No. 85, April 1967, pp. 178-182

Davis J.A., "A Sheet Model of the Beam-Plasma Interaction" American Physical Society Meeting, Division of Plasma Physics, Boston, Mass., November 2-5, 1966

Davis, J.A., "Computer Simulation of the Beam-Plasma Interaction", Symposium on Computer Simulation of Plasmas and Many Body Problems, Williamsburg, Va., April 19-21, 1967

Denning, P.J., "Effects of Scheduling on File Memory Operations", AFIPS Proceedings of the Spring Joint Computer Conference, 1967, Thompson Book Company, Washington, D.C., pp. 14-22

Dertouzos, M.L., and H.L. Graham, "A Parametric Graphical Display Technique for On-Line Use", AFIPS Proceedings of the Fall Joint Computer Conference, 1966

Dertouzos, M.L., and Z.C. Fluhr, "Minimization and Convexity in Threshold Logic", Symposium on Switching and Automata Theory, Proceedings of the 7th Annual Meeting, Berkeley, California, October 1966

Fano, R.M., "The Computer Utility and the Community", IEEE Convention Record, March 20-23, 1967

Fano, R.M., and F.J. Corbató, "Time-Sharing on Computers", Scientific American, vol. 215, no. 3, September 1966

Frey, F.W., "Regional Variations in Rural Turkey", Rural Development Research Project, sponsored jointly by the Government of Turkey and the United States Agency for International Development, December 1966

Frey, F.W., "Socialization to National Identification: Turkish Peasants", paper delivered to American Political Science Convention, September 1966

Greenberger, M., "Computers in Business and Government", Encyclopedia Americana, 1967

Greenberger, M., "The Priority and Computer Time-Sharing", Management Science, July 1966

Greenberger, M., "The Uses of Computers in Organizations", Scientific American, vol. 215, no. 3, September 1966 (reprinted in Information, Freeman, 1966)

Greenberger, M., "The Urban Node in the Information Network", Proceedings of the IEEE National Convention, March 20, 1967

Greenberger, M., "Computer in the Organization", Modern Technology and Civilization, Charles R. Walker (editor)

Greenberger, M., and M.M. Jones, "On-Line Simulation in the OPS System", Proceedings of the 21st ACM National Conference, August 1966

Greenberger, M., and M.M. Jones "On-Line Incremental Simulation", Proceedings IFIP Working Conference on Simulation Languages, Oslo, Norway, May 1967

Griffei, D., and S. McIntosh, "The Current ADMINS System", Project Intrex Seminar, January 1967

Gronemann, U.F., Knudson, D.R., and S.N. Teicher, Remote Text Access for Project Intrex, M.I.T. E.S.L Technical Report ESL-TM-312, July 1967

Groneman, U.F., Operation Manual for The Coditcon: A TV Scan-Conversion and Image-Storage System for Computer Displays, M.I.T. E.S.L. Report R-284, July 1966

Gustafson, T.K., and H.A. Hause, "Nonlinear Amplification", M.I.T. R.L.E. Quarterly Progress Report No. 84, January 1967, pp. 113-120

Guzmán, A., and H.V. McIntosh, "Comments on 'All Paths Through a Maze' ", Proceedings of the IEEE, (letters)

Guzmán, A., and H.V. McIntosh, "CONVERT", Communications of the ACM 9, 8: 604-615, August 1966

Hansen, K.F., "Comparative Analysis of Problems in Nuclear Fuel Burnup, Reactor Kinetics and Dynamics as a Function of Computer Capability", International Conference on Reactor Mathematics and Computation, Mexico City, May 1967

Haring, D. R., "A Terminal for an On-Line Interactive Retrieval System", IEEE Workshop on Advanced Computer Peripherals, Lake Arrowhead, California, August 25-27, 1967

Haring, D., "Multi-Threshold Building Blocks", IEEE Transaction on Electronic Computers, vol. EC-15, no. 4, August 1966

Haus, H.A., and A.L. McWhorter, "Amplification at Subcritical Drift Velocities", M.I.T. R.L.E. Quarterly Progress Report No. 82, July 1966, p. 91

Henke, W. L., "Dynamic Articulatory Model of Speech Production", Acoustical Society of America, Los Angeles, California, presented at the 72nd meeting, November 1966

Ingraham, J.C., "Ion-Wave Instability in a Steady-State Discharge", M.I.T. R.L.E. Quarterly Progress Report No. 82, October 1966, pp. 37-40

Johnson, T.J.R., "A Branch and Bound Algorithm for The Resource Constrained Project Scheduling Problem", presented at the National Meeting of The Institute for Management Sciences (TIMS), April 1967 (paper)

Kaplow, R., Brackett, J.W. and S.L. Strong, "Man-Machine Communication in On-Line Mathematical Analysis", AFIPS Proceedings of Joint Computer Conference, November 1966

Katzenelson, J., and L.H. Seitman, "An Iterative Method for Solution of Nonlinear Monotone Resistors", IEEE Transactions on Circuit Theory, vol. CT-13, no. 3, September 1966

Katzenelson, J., "AEDNET: A Simulator for Nonlinear Networks", Proceedings of the IEEE, vol. 54, no. 1, November 1966

Kendrick, D.A., and M. Weitzman, Progress Report on Branch and Bound Algorithm for Zero-One Mixed Integer Programming Problems, Center for International Studies, June 1966

Kohavi Z., and P. Lavallee, "Design of Diagnosable Sequential Machines", Proceedings of the 1967 Spring Joint Computer Conference, vol. 30, pp. 713-718

Krakauer, L., "PSYCHLOPS: A System for Using the PDP-4 Computer for On-Line Adaptive Psychophysical Experiments", M.I.T. R.L.E. Quarterly Progress Report No. 82, July 1966, pp. 252-253

Kusse, B.R., "Stability Criteria for Dispersion Relations Containing Branch Points", M.I.T. R.L.E. Quarterly Progress Report No. 84, January 1967, pp. 151-154

Kusse, B.R., and A. Bers, "Cross-Field Beam-Plasma Experiment", M.I.T. R.L.E. Quarterly Progress Report No. 84, January 1967, p. 144

Kusse, B.R., and A. Bers, "Cross-Field Beam-Plasma Interactions", M.I.T. R.L.E. Quarterly Progress Report No. 82, July 1966, pp. 154-157

Landin, P.J., A λ-Calculus Approach, Advances in Programming and Non-Numerical Computation, Pergamon Press, 1966

Lee, F.F., Troxel, D.E., and S.J. Mason, "Research Objectives and Summary of Research", M.I.T. R.L.E. Quarterly Progress Report No. 84, January 1967, pp. 287-289

Lerner, D., "Interviewing European Elites", Polls, vol. II, no. 1, Fall 1966

Lieberman, M.A., and A. Bers, "Ion Oscillations Excited by Electron Beam-Plasma Interactions", presented at the American Physical Society Meeting, Division of Plasma Physics, Boston, Mass., November 2-5, 1966

Lieberman, M.A., and A. Bers, "Quasilinear Theory of Narrow Bandwidth Convective Instabilities", presented at the American Physical Society Meeting, Division of Plasma Physics, Boston, Mass., November 2-5, 1966

Little, J.D.C., "A Model of Adaptive Control of Promotional Spending", Operations Research, Vol. 14, pp. 1075-1098, November 1966

Little, J.D.C., "The Synchronization of Traffic Signals by Mixed-Integer Linear Programming", Operations Research, vol. 14, July 1966

Llewellyn-Jones, D.T., and E.M. Mattison, "Electromechanical Device to Feed Experimental Data Automatically into a Time-Shared Computer System", M.I.T. R.L.E. Quarterly Progress Report No. 82, July 1966, pp. 93-97

McLoud, V.E., and M.M. Pennell, "Example of Symbolic Manipulation of Polynomials in MAD", M.I.T. R.L.E. Quarterly Progress Report No. 82, July 1966, pp. 294-295

Manove, M., S. Bloom and C. Engelman, Rational Functions in MATHLAB, Mitre Corporation, Technical Report MTP-35, August 1966

Miller, J.R., "On-Line Analytical Routines for Social Science Research", International Symposium on Mathematical and Computational Methods in Sciences, Rome, July 1966

Miller, J.R., On-Line Analysis for Social Scientists, M.I.T. Sloan School Working Paper 226-66, November 1966

Miller, R.L., A Model for Traffic Flows on a Two-Lane Two-Way Rural Highway, Operations Research Center, Report TR-20, June 1966

Mills, R.G., "Computers and Communications - Their System Interactions", IEEE Symposium Proceedings, Santa Monica, California, January 19, 1967

Minsky, M.L., "Artificial Intelligence", Scientific American, vol. 215, no. 3, September 1966

Ness, D., and H.M. Weingartner, Methods for the Solution of the Multi-Dimensional 0/1 Knapsack Problem, Sloan School of Management, May 1966; also published in Operations Research, May 1966

Ness, D., and H.M. Weingartner, "Computational Methods for the Multi-Dimensional 0/1 Knapsack Problem", Operations Research, January-February 1967

Oettinger, A.G., "The Use of Computers in Science", Scientific American, vol. 215, no. 3, September 1966

Overhage, C.F.J., "Project Intrex — Plans and Progress", Harvard University Librarian's Seminar, Cambridge, Mass., May 17, 1967

Overhage, C.F.J., "Science Libraries: Prospects and Problems", Science, 155, February 1967, pp. 802-806

Parker, R. and A. Bers, "Excitation and Propagation of Ion-Cyclotron Waves in a Bounded Plasma", presented at the American Physical Society Meeting, Division of Plasma Physics, Boston, Mass., November 2-5, 1966

Pennell, M.M., "Research Objectives", M.I.T. R.L.E. Quarterly Progress Report No. 84, January 1967, p. 349

Pennell, M.M., and H. Davis, "Linearizing the Roots of a Polynomial", M.I.T. R.L.E. Quarterly Progress Report No. 83, October 1966, pp. 179-181

Pennell, M.M., and J. Harwitt, "Roots, A Root-Finding Subroutine Using Muller's Method", M.I.T. R.L.E. Quarterly Progress Report No. 84, January 1967, pp. 352-354

Pennell, M.M., and R.M. Nacamuli, "A Computer Indexing Program", M.I.T. R.L.E. Quarterly Progress Report No. 82, July 1966, pp. 293-294

Quarton, G.C., M.T. McGuire, and S. Lorch, "Man-Machine Natural Language Exchanges Based on Selected Features of Unrestricted Input", (mimeo, 1966), The Stanley Cobb Laboratory for Psychiatric Research, Massachusetts General Hospital and Harvard Medical School

Reilly, R.D., and W.D. Getty, "Spectrographic Measurement of Electron Temperature in the Beam-Plasma Discharge", M.I.T. R.L.E. Quarterly Progress Report No. 82, July 1966, pp. 144-145

Reintjes, J.F., and C.H. Perry, "Low-Frequency Vibrations in Ammonium-Chloride and Ammonium-Bromide Crystals", M.I.T. R.L.E. Quarterly Progress Report No. 82, July 1966, pp. 48-53

River, E.C., "Finding the Roots of an Analytic Function", M.I.T. R.L.E. Quarterly Progress Report No. 85, April 1967, p. 329

River, E.C., "A Simple Method for Finding the Roots of an Analytic Function", M.I.T. R.L.E. Quarterly Progress Report No. 84, January 1967, pp. 351-352

River, E.C., and N. Benhaim, "Real-Time Spectral Input System for Computer Analysis of Speech", M.I.T. R.L.E. Quarterly Progress Report No. 84, January 1967, pp. 253-254

Rosenberg, R.C., Computer-Aided Teaching of Dynamic System Behavior, Electrical System Division, Technical Report ESD-TR-66-260, Hanscom Field, Mass., November 1965

Rosenberg, R.C., "The Dynamic Systems Laboratory", ASEE Annual Meeting, June 1966

Ross, D.T., S.A. Coons and J.E. Ward, "Investigations in Computer-Aided Design for Numerically Controlled Production", M.I.T. E.S.L. Interim Engineering Progress Report, June 1965 through May 1966, ESL-IR-278, August 1966

Ruyle, A., J.W. Brackett and R. Kaplow, "A Survey of Systems for On-Line Mathematical Assistance", Proceedings of the ACM National Conference, 1967

Schneider, H.M., "Dynamics of the Plasma Boundary", M.I.T. R.L.E. Quarterly Progress Report No. 82, July 1966, pp. 157-161

Schneider, H.M., "Sheet Model of a Plasma Slab", M.I.T. R.L.E. Quarterly Progress Report No. 84, January 1967, pp. 149-151

Schneider, H.M., and A. Bers, "Nonlinear Effects in Plasma Slab Oscillations", presented at the Computer Simulation Symposium, College of William and Mary, Williamsburg, Va., April 19, 1967

Skinner, Thomas P., "The Computer Display in a Time-Sharing Environment", Digital Equipment Computer Users Society Spring Symposium, April 14, 15, 1967

Smullin, L.D., A. Bers, R.J. Briggs and R.R. Parker, "Research Objectives", M.I.T. R.L.E. Quarterly Progress Report No. 84, January 1967, p. 141

Speck, C.E., and A. Bers, "Observation of Enhanced Cyclotron Radiation from an Electron-Cyclotron Resonance Discharge", M.I.T. R.L.E. Quarterly Progress Report No. 82, July 1966, pp. 187-189

Stevens, C.H., "New Developments in Libraries", State University of New York (Albany), April 4, 1967

Stotz, R.H., and T.B. Cheek, "A Low-Cost Graphic Display for a Computer Time-Sharing Console", 8th Annual National Symposium on Information Display, San Francisco, May 1967

Weingartner, H.M., "Capital Budgeting of Interrelated Projects: Survey and Synthesis", Management Science, Series A - Sciences, March 1966

Winiecki, K. B., TACT On-Line Computing System, Version II, Harvard University Computation Laboratory, December 1966

Weizenbaum, J., "ELIZA: A Computer Program for the Study of Natural Communication between Man and Machine", Communications of the A. C. M., 1966, pp. 36-45

APPENDIX D

PROJECT MAC TECHNICAL REPORTS

REPORT NOS.	DDC NOS.	TITLE	AUTHOR(S)	DATE
MAC-TR-1 (THESIS)	AD-604-730	Natural Language Input for a Computer Problem Solving Language	Bobrow, D. G.	6/64
MAC-TR-2 (THESIS)	AD-608-499	SIR: A Computer Program for Semantic Information Retrieval	Raphael, B.	6/64
MAC-TR-3	AD-608-501	System Requirements for Multiple-Access, Time-Shared Computers	Corbató, F. J.	5/64
MAC-TR-4	AD-604-678	Verbal and Graphical Language for the AED System: A Progress Report	Ross, D. T. Feldman, C. G.	5/64
MAC-TR-6	AD-605-679	STRESS: A Problem-Oriented Language for Structural Engineering	Biggs, J. M. Logeher, R. D.	5/64
MAC-TR-7	AD-604-680	OPL-1: An Open-Ended Programming System within CTSS	Weizenbaum, J.	4/64
MAC-TR-8	AD-604-681	The OPS-1 Manual	Greenberger, M.	5/64
MAC-TR-11	AD-608-500	Program Structure in a Multi-Access Computer	Dennis, J. B.	5/64
MAC-TR-12	AD-609-296	The MAC System: A Progress Report	Fano, R. M.	6/64
MAC-TR-13	AD-609-288	A New Methodology for Computer Simulation	Greenberger, M.	10/64
MAC-TR-14	AD-661-807	Use of CTSS in a Teaching Environment	Roos, D.	11/64
MAC-TR-16	AD-612-702	CTSS Technical Notes	Saltzer, J. H.	3/65
MAC-TR-17	AD-462-158	Time-Sharing on a Multi-Console Computer	Samuel, A. L.	3/65
MAC-TR-18	AD-470-715	An Analysis of Time-Shared Computer Systems	Scherr, A. L.	6/65

REPORT NOS.	DDC NOS.	TITLE	AUTHOR(S)	DATE
MAC-TR-19 (THESIS)	AD-474-018	A Heuristic Approach to Alternate Routing in a Job Shop	Russo, F. J.	6/65
MAC-TR-20 (THESIS)	AD-474-019	CALCULAID: An On-Line System for Algebraic Computation and Analysis	Wantman, M. E.	9/65
MAC-TR-21 (THESIS)	AD-624-943	Queueing Models for File Memory Operation	Denning, P. J.	10/65
MAC-TR-22	AD-625-728	The Priority Problem	Greenberger, M.	11/65
MAC-TR-23	AD-627-537	Programming Semantics for Multiprogrammed Computations	Dennis, J. B. Van Horn, E. C.	12/65
MAC-TR-24	AD-476-443	MAP: A System for On-Line Mathematical Analysis	Kaplow, R. Strong, S. L. Brackett, J. W.	1/66
MAC-TR-25 (THESIS)	AD-631-396	Investigation of an Analog Technique to Decrease Pen-Tracking Time in Computer Displays	Stratton, W. D.	3/66
MAC-TR-26 (THESIS)	AD-631-269	Design of a Low-Cost Character Generator for Remote Computer Displays	Cheek, T. B.	3/66
MAC-TR-27 (THESIS)	AD-633-678	OCAS: On-Line Cryptanalytic Aid System	Edwards, D. J.	5/66
MAC-TR-28 (THESIS)	AD-637-215	Input/Output in Time-Shared, Segmented, Multiprocessor Systems	Smith, A. A.	6/66
MAC-TR-29	AD-636-275	Search Procedures Based on Measures of Relatedness Between Documents	Ivie, E. L.	6/66
MAC-TR-30 (THESIS)	AD-635-966	Traffic Control in a Multiplexed Computer System	Saltzer, J. H.	7/66
MAC-TR-31 (THESIS)	AD-637-192	Models and Data Structures for Digital Logic Simulation	Smith, D. L.	8/66
MAC-TR-32 (THESIS)	AD-638-446	PILOT: A Step Toward Man-Computer Symbiosis	Teitelman, W.	9/66

REPORT NOS.	DDC NOS.	TITLE	AUTHOR(S)	DATE
MAC-TR-33 (THESIS)	AD-645-660	ADEPT: A Heuristic Program for Proving Theorems of Group Theory	Norton, L. M.	10/66
MAC-TR-34 (THESIS)	AD-650-407	Computer Design for Asynchronously Reproducible Multiprocessing	Van Horn, E. C.	11/66
MAC-TR-35 (THESIS)	AD-657-282	An On-Line System for Algebraic Manipulation	Fenichel, R. R.	12/66
MAC-TR-36 (THESIS)	AD-657-283	Symbolic Mathematical Laboratory	Martin, W. A.	1/67
MAC-TR-37 (THESIS)	AD-656-041	Some Aspects of Pattern Recognition by Computer	Guzman, A.	2/67
MAC-TR-38	AD-662-027	A Low-Cost Output Terminal	Rosenberg, R. Kennedy, D. W. Humphrey, R. A.	3/66
MAC-TR-39	AD-661-806	Syntax-Based Analytic Reading of Musical Scores	Forte, A.	4/67
MAC-TR-40	AD-668-009	On-Line Analysis for Social Scientists	Miller, J. R.	5/67
MAC-TR-41	AD-663-504	Surfaces for Computer-Aided Design of Space Forms	Coons, S. A.	6/67
MAC-PR-1	AD-465-088	Progress to July 1964	Fano, R. M., et al	
MAC-PR-2	AD-629-494	Progress Report II July 1964 to July 1965	Fano, R. M., et al	
MAC-PR-3	AD-648-346	Progress Report III July 1965 to July 1966	Fano, R. M., et al	

AUTHORS INDEX

ADMINISTRATION

ARTIFICIAL INTELLIGENCE

BIOLOGY DEPARTMENT

CIVIL ENGINEERING DEPARTMENT

COMPUTATION STRUCTURES

COMPUTER SYSTEM RESEARCH

ELECTRONIC SYSTEMS LABORATORY

HARVARD UNIVERSITY

MAN-MACHINE COMMUNICATION

PROGRAMMING LINGUISTICS

RESEARCH LABORATORY OF ELECTRONICS

SCHOOL OF ENGINEERING

SCHOOL OF HUMANITIES AND SOCIAL SCIENCES

SCHOOL OF SCIENCE

SLOAN SCHOOL OF MANAGEMENT

TECHNICAL INFORMATION PROGRAM

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Massachusetts Institute of Technology Project MAC	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP None
3. REPORT TITLE Project MAC Progress Report IV July 1966 to July 1967	
4. DESCRIPTIVE NOTES (Type of report and inclusive dates) annual progress	
5. AUTHOR(S) (Last name, first name, initial) Collection of material from many Project MAC participants; R. M. Fano, Director	
6. REPORT DATE 1 July 1967	7a. TOTAL NO. OF PAGES 282 7b. NO. OF REFS ——
8a. CONTRACT OR GRANT NO. Office of Naval Research, Nonr-4102(01) b. PROJECT NO. NR-048-189 c. RR 003-09-01 d.	9a. ORIGINATOR'S REPORT NUMBER(S) MAC-PR-4 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
10. AVAILABILITY/LIMITATION NOTICES Distribution of this document is unlimited.	
11. SUPPLEMENTARY NOTES None	12. SPONSORING MILITARY ACTIVITY Advanced Research Projects Agency 3D-200 Pentagon Washington, D.C. 20301

13. ABSTRACT

The broad goal of Project MAC is experimental investigation of new ways in which on-line use of computers can aid people in their individual work, whether research, engineering design, management, or education.

This is the fourth annual Progress Report summarizing the research carried out under the sponsorship of Project MAC. Details of this research may be found in the publications listed in the Appendices at the end of this report.

14. KEY WORDS

Computers	On-line computers	Time-shared computers
Machine-aided cognition	Real-time computers	
Multiple-access computers	Time-sharing	

DD FORM 1 NOV 65 1473 (M.I.T.)

UNCLASSIFIED

Security Classification